TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PANU PARVIAINEN

MATLAB-OPPIMATERIAALIN KEHITTÄMINEN TAMPEREEN TEKNILLISESSÄ YLIOPISTOSSA

Diplomityö

Tarkastaja: Simo Ali-Löytty
Tarkastaja ja aihe hyväksytty
Luonnontieteiden tiedekuntaneuvoston
kokouksessa 12.8.2015

TIIVISTELMÄ

PANU PARVIAINEN: MATLAB-oppimateriaalin kehittäminen Tampereen teknillisessä yliopistossa
Tampereen teknillinen yliopisto
Diplomityö, 57 sivua, 42 liitesivua
tammikuu 2016
Teknis-luonnontieteellinen koulutusohjelma
Pääaine: Matematiikka
Tarkastaja: Yliopistonlehtori Simo Ali-Löytty
Avainsanat: E-oppiminen, MATLAB, Moodle, STACK, klusterianalyysi

Syksyllä 2015 Tampereen teknillisessä yliopistossa pilotoitiin uutta e-oppimateriaalia hyödyntävää MATLAB-kokonaisuutta. Matematiikan laitoksen tehtävänä oli kehittää e-oppimateriaalien avulla ensimmäisen vuosikurssin opiskelijoille pakollista MATLABin alkeiden kokonaisuutta, jonka tavoitteena oli perehdyttää uudet opiskelijat MATLABin perustoiminnallisuuksiin. MATLABin alkeiden kokonaisuuden kehittämiseen otettiin vaikutteita e-oppimisen pedagogiikasta ja muista Suomen tai muiden maiden korkeakoulujen käyttämistä e-oppimateriaalipainotteisista laskentaohjelmaopetuskokonaisuuksista.

TTY:n MATLABin alkeiden kokonaisuus toteutettiin Moodle-sivuna, jonne koottiin itse tehty e-oppimateriaali. Oppimateriaali koostui automaattisesti itsensä tarkastavista STACK-tehtävistä, opetusvideoista ja ohjepdf-tiedostoista. STACK-tehtävät olivat arvioitava osuus, jotka jokaisen opiskelijan tuli palauttaa. Tehtävien tueksi oli opetusvideot ja ohjepdf-tiedostot, joilla käsiteltiin tehtävien ratkaisemiseen liittyvää materiaalia mutta myös ylimääräistäkin MATLAB-tietoutta. Moodle-sivulta löytyi myös keskustelupalsta opiskelijoiden ja opettajien välille sekä toisen periodin ensimmäisen viikon aikana järjestettiin ohjattuja harjoitustilaisuuksia, joissa opiskelijat voivat ratkoa tehtäviä.

Alkeiden kokonaisuuden kokeilu onnistui pääosin hyvin, sillä suurin osa noin 700 opiskelijasta läpäisi alkeet, ja e-oppimateriaali koettiin käyttökelpoiseksi työkaluksi. Opiskelijoilta saadun palautteen ja opiskelijoiden suorituskäyttäytymisestä tehdyn klusterianalyysin perusteella alkeiden kokonaisuutta voi sisällöllisesti jatkossa kehittää entistä enemmän MATLABin käyttötaitojen oppimista ja sisäistettävyytä edistäväksi kokonaisuudeksi.

ABSTRACT

PANU PARVIAINEN: Developing MATLAB based Learning Material in Tampere University of Technology
Tampere University of Technology
Master's thesis, 57 pages, 42 Appendix pages
January 2016
Master's Degree Programme in Science and Engineering
Major: Mathematics
Examiner: University Lecturer Simo Ali-Löytty
Keywords: E-learning, MATLAB, Moodle, STACK, cluster analysis

A new pilot project regarding the development of MATLAB based e-learning material was conducted in the fall 2015 at Tampere University of Technology. The department of Mathematics at TUT was to develop a compulsory teaching module of introduction to MATLAB for first year students. This module was to consist mainly of e-learning material and it would introduce new university students into the basic functionality of MATLAB. The pedagogical methods of e-learning and other e-learning material based computational software courses in Finnish universities and elsewhere around the world served as inspiration in developing this new introductory material.

A Moodle course for Introduction to MATLAB in TUT was made where the self made e-learning material was gathered. The material consisted of automatically self checking STACK assignments, teaching videos, and pdf files. The STACK assignments were the evaluated part of the course, and every student had to return the assignments in due time. The videos and the pdf files contained supporting information for solving the assignments and also extra knowledge of MATLAB. There was also a forum in the Moodle course where students and teachers could communicate with each other, and during the first lecture week of the second period in TUT exercise sessions were organised where students could solve the assignments under the guidance of a teaching assistant.

The introductory course succeeded mainly well, for most of all the 700 students passed the introductory MATLAB course. The e-learning material was also regarded as a useful tool. Due to the feedback received from the students and the cluster analysis made to study the behavioural patterns of the students it can be concluded that the composition of the introductory course can be developed even further.

ALKUSANAT

Tämä diplomityö on tehty erään opiskelijan valmistumista silmällä pitäen ja sen tarkoituksena on myös avittaa muiden opiskelijoiden opintoihin keskittymistä MATLAB-opetuksen muodossa. Opinnäytetyö on tehty TTY:n matematiikan laitokselle, ja työhön ovat jonkin verran osallistuneet kaikki syksyn 2015 MATLAB-pilotin osalliset tahot. Diplomityöprosessin on ohjannut Simo Ali-Löytty.

Haluan erityisen paljon kiittää ohjaaja-Simoa hänen tarjoamastaan tuesta paitsi diplomityöprosessin myös kandidaatintyöprosessini osalta. Lisäksi haluan kiittää muuta matematiikan laitoksen henkilökuntaa ja erityisesti samassa laitoksen työhuoneessa kanssani työskennelleitä lyhytaikaisen sijoitukseni ajalta. Teidän neuvoistanne oli hyötyä. Lopuksi haluan vaatimattomasti kiittää kaikkia ihmisiä, joiden kanssa olen koko opiskeluni ja sitä myötä myös diplomityöprosessini aikana ollut tekemisissä. Te mahdollistitte sen, että perustutkinto-opiskelu oli jotain täysin sanoinkuvaamatonta.

Jatka lukemista ja koe, millaista on kuuden ja puolen vuoden opiskelun anti.

Tampereella 27. joulukuuta 2015

Panu Parviainen

SISÄLLYS

KUVALUETTELO

TAULUKKOLUETTELO

LYHENTEET JA MERKINNÄT

E-materiaali	Opetusmateriaali, joka hyödyntää teknologiaa, esimerkiksi tietokoneavusteinen oppimateriaali.
MATLAB	Tulee sanoista Matrix Laboratory. Yleinen insinöörien ja tiedemiesten käytössä oleva matriisilaskentaan perustuva laskentaohjelmisto.
MOOC	Tulee sanoista Massive Online Open Course. Kaikille avoin verkkokurssitoteutus.
STACK	Tulee sanoista System for Teaching and Assessment using a Computer algebra Kernel. Järjestelmä, jolla voi luoda Moodle-tenttiin matematiikkaa hyödyntäviä automaattisesti itsensä tarkastavia tehtäviä.
MIT	Tulee sanoista Massachusetts Institute of Technology. Yhdysvaltalainen teknillinen korkeakoulu.
CAS	Tulee sanoista Computer Algebra System. Järjestelmä, joka käyttää tietokonealgebraa.
rand	Lyhenne sanasta random. Tarkoittaa satunnaista lukuarvoa, joka voi olla käyttöyhteydestä riippuen eri tavalla satunnainen.
ESS	Tulee sanoista Error Sum of Squares eli virheiden neliösumma.

$A \otimes B$	Matriisien A ja B välinen Kroneckerin tulo
$\lfloor a \rfloor$	Luvun a lattia eli suurin kokonaisluku, joka ei ole suurempi kuin a
$det(A)$	Matriisin A determinantti
$\mathbf{x}$	Satunnaisvektori
d	Dissimilaarisuusmitta
D	Diskriminantti tai dissimilaarisuusmatriisi
$\bar{\mathbf{x}}$	Klusterin painopiste

1. JOHDANTO

MATLAB on pitkään ollut Tampereen teknillisellä yliopistolla tuttu laskentaohjelmisto esimerkiksi numeeriselle laskennalle. MATLAB on myös yleisin laskentaohjelmisto TTY:n matematiikan, fysiikan, systeemitekniikan ja signaalinkäsittelyn laitoksilla. Maailmanlaajuisestikin MATLAB on hyvin yleinen insinöörien ja tiedemisten käyttämä laskentaohjelmisto.

MATLABin käyttöä on TTY:n sopiskelijoille opetettu enemmän tai vähemmän menestyksekkäästi useana vuotena. Oppimateriaalin uudistuksen halusta sekä olosuhteiden pakosta syksyllä 2015 lähdettiin pilotoimaan MATLABin oppimateriaalikokonaisuutta, johon osallistuivat ne TTY:n laitokset, jotka MATLABia enimmäkseen käyttävät ja jotka arvelevat uudistuneesta MATLAB-materiaalista olevan hyötyä. Pilotin tarkoituksena oli tehdä merkittävämmäksi e-oppimateriaalin osuutta MATLAB-oppimateriaalissa entisen kontaktiopetuspainotteisen opetuksen sijaan. Kukin osallinen laitos lähti toteuttamaan pilottia omalla tavallaan. Matematiikan laitos sai tehtäväkseen uudistaa ensimmäisen vuosikurssin opiskelijoille pakollisen MATLABin alkeiden kokonaisuuden, jonka on tarkoitus perehdyttää uudet opiskelijat MATLABin perustoiminnallisuuksiin. Tässä työssä tarkastellaan enimmäkseen näitä MATLABin alkeita, niiden taustoja, menetelmiä ja tuloksia.

Työ alkaa katsauksella e-oppimisen ja e-oppimateriaalien taustalla olevaan pedagogiikkaan. Tätä taustaa hyödynnetään seuraavissa luvuissa, kun tarkastellaan ensin muun muassa muiden e-oppimista hyödyntävien laskentaohjelmistojen oppimateriaalia ja niiden taustalla olevaa pedagogiikkaa. Oppimateriaalia on haettu niin Suomesta kuin ulkomailtakin, ja luonnollisesti ulkopuolisen materiaalin tarkastelu painottuu MATLAB-oppimateriaaleihin. Sen jälkeen käsitellään yksityiskohtaisesti TTY:llä pilotoidun MATLABin alkeiden oppimateriaalikokonaisuutta ja erityisesti arvioinnissa keskeisessä osassa olleita STACK-tehtäviä. Alkeiden oppimateriaalivalinnassa tehdyt ratkaisut perustellaan pedagogisella taustalla ja muilla laskentaohjelmistojen opetukseen keskittyvällä e-oppimateriaalilla. Lopuksi tarkastellaan alkeiden kokonaisuuden toteutusta ja toteutuksen onnistumista. Opiskelijoiden suoritusaktiivisuudesta tehdään myös tilastollista klusterianalyysiä, jolla pyritään kartoittamaan harjoitustentekokäyttäytymistä kehityskohteita ajatellen.

2. E-OPPIMINEN

2.1 Mitä e-oppiminen on?

E-oppimisen määrittely ei ole nykypäivänä täysin yksiselitteinen. Digitalisaation ja teknologian kehityksen myötä erityyppisten e-oppimateriaalien kirjokin on kasvanut merkittävän kattavaksi. E-opetus mielletään kuitenkin useimmiten teknologisavusteiseksi opettamiseksi. Suurimmassa osassa tapauksia e-oppiminen vieläpä ensisijaisesti tarkoittaa juurikin sellaista opetuksen muotoa, jossa niin sanottu perinteinen opetustyyli on syrjäytetty. E-oppiminen ei siis suoraan tarkoita sitä, että perinteisellä oppitunnilla käytetään oppimismetodeina tavallista enemmän teknisiä apuvälineitä. Sen sijaan e-oppimisen tyylien mukaista on esimerkiksi itsenäisen oppimisen mahdollistavat verkkoympäristöt. Liisa Ilomäen ja opetushallituksen laatimassa kirjassa nimeltä Laatua e-oppimateriaaleihin - e-oppimateriaalit opetuksessa ja oppimisessa mainitaan opetushallituksen määrittelemäksi e-oppimateriaaliksi kaikki sellainen oppimateriaali, joka on verkossa saatavilla oleva oppimateriaaliksi tarkoitettu sisältö [1, s. 5].

E-oppimisen sijaan voidaan puhua myös verkko-oppimisesta. Löfströmin, Kanervan, Tuuttilan ja Nevgin teos Laadukkaasti verkossa - Verkko-opetuksen käsikirja yliopisto-opettajalle käsittelee nimenomaisesti käsitteen verkko-oppiminen erityispiirteitä ja haasteita. Käsikirja keskittyy ensisijaisesti yliopisto-opettajan näkökulmaan asiassa, mutta samoja ideoita voi todennäköisesti hieman muunnellen soveltaa myös muillekin kouluasteille. Tässäkin teoksessa mainitaan erityisesti, että verkko-oppimisessa opettajan ja opiskelijan välinen vuorovaikutus on selkeästi erilaista kuin kasvokkain tapahtuvassa kontaktiopetuksessa. Tämän vuoksi verkko-oppimisessa opetuksen painopisteet on mietittävä eri tavoin perinteiseen opetukseen nähden. Iso osa verkko-opetuksen onnistumisesta onkin perustavanlaatuisessa ja perinpohjaisessa suunnittelussa. Kirjan mukaan termillä verkko-opetus tarkoitetaan sulautettua opetusta, jossa kasvokkain ja verkkoympäristössä tapahtuva opetus ja oppiminen yhdistyvät. [2]

Ilomäen ja opetushallituksen kirja antaa luokittelun erityyppisestä e-oppimateriaalista. Erilaisten e-oppimateriaalien ymmärtämiseksi luokittelu on yleensä hyvä ratkaisu toteuttaa. Ilomäen kirjan mukaan luokittelun on tehnyt opetushallitus. Luokittelu menee seuraavasti: arviointi, avoin toiminta, blogi, demonstraatio, esitys, harjoitusohjelmat, kurssi, opas, oppimispeli, simulaatio tai mallinnusohjelma, tietolähde, tutkivan oppimisen tueksi tehty materiaali, työkalu ja wiki. Tämä luokittelu on luonnollisesti vain yksi tapa jäsentää e-oppimateriaalia. Luokittelu on kuitenkin toimiva, eivätkä muut luokittelut merkittävästi eroa tästä. [1, s. 8-9]

Opetushallituksen luokittelussa kuvataan tarkemmin äsken esiteltyjä käsitteitä. Arvioinnilla tarkoitetaan sitä kun oppijan osaamista arvioidaan tai hän arvioi sitä itse. Tähän kuuluvat sähköiset ja koneellistetut arvioinnit, kuten monivalinta-, aukko- ja laskutehtävät sekä sähköiset kokeet. Esimerkiksi oppimisalusta Moodle tarjoaa oppijalle sekä itsearviointiin että ylhäältä tulevaan arviointiin mahdollistetut toiminnot. Hieman äskeistä vapaampaa formaattia käsittelee avoin toiminta, joka pitää sisällään avoimia tehtäviä tai luovia harjoituksia, jolloin oppijoiden toiminnat ja toimintojen tulokset eivät ole etukäteen ennustettavissa. Blogi, joka on vapaasti suomennettuna verkkoloki tai verkkopäiväkirja, sisältää opetussuunitelmien toteutuksen kannalta merkittävää aineistoa. Demonstraatio on vastaava kuin esimerkiksi opettajajohtoinen kemiallinen koe kemian oppitunnilla mutta verkkototeutusmuotoisena. Demonstraation ideana on esitellä ilmiöitä tai asioita, joiden parissa käyttäjä ei itse voi työskennellä. Esityksellä tarkoitetaan yksinkertaisesti jotain sähköistä esitysformaattia, jossa opiskeltavat yksityiskohdat esitetään tiivistetyssä muodossa pedagogisesti mielekkäällä tavalla. Tällaisia esityksiä voivat olla esimerkiksi diaesitykset tai videot. Harjoitusohjelmia voi olla monenlaisia ja niiden päämääränä on harjoitella aiemmin opetettuja asioita. Kurssilla tarkoitetaan tässä yhteydessä ensisijaisesti verkkokurssia, jonka oppija voi usein suorittaa itsenäisesti vapaaseen tahtiin. Opas on tässä yhteydessä sellainen toiminne, joka ohjaa käyttäjää jonkin konkreettisen toiminnan suorittamiseen ja se voi sisältää myös toiminnan tai asian havainnollistamista esimerkiksi tekstein, kuvin, äänin ja animaatioin. Oppimispeli on yksinkertaisesti pelimuotoon rakennettu oppimistapahtuma. Simulaation tai mallinnusohjelman avulla oppija pääsee itse osallistumaan jonkin kuvitteellisen tai todellisen prosessin jäljittelemiseen ja mallintamiseen tavoitteenaan tietojen ja taitojen oppiminen ja soveltaminen. Tietolähteellä tarkoitetaan jotain muun oppimateriaalin ulkopuolista aiheeseen liittyvää aineistoa, jota ei välttämättä ole jäsennetty pedagogisten tavoitteiden mukaisesti. Tutkivan oppimisen tueksi tehty materiaali ohjaa oppijaa tekemällä oppimiseen ja kehittämään ongelmanratkaisutaitoja. Työkalulla tarkoitetaan tässä yhteydessä sovellusta, jolla oppija voi tuottaa jotain uutta, ilmaista itseään, muokata aiemmin laadittua materiaalia tai olla vuorovaikutuksessa toisten kanssa. Wikit ovat viime aikoina kasvattaneet merkitystään ja tässä tapauksessa e-oppimateriaaliwikissä on ainoastaan opetussuunnitelmien toteutuksen kannalta merkittävää aineistoa. [1, s. 8-9]

Tästä luokittelusta huomataankin, että suurin osa e-oppimateriaaleista on idealtaan juuri sellaisia, että ne tarjoavat oppilaalle tai opiskelijalle internet-yhteyden avulla oppimateriaalia itsenäistä opiskelua varten. Luettelosta nähdään myös, että jo tässä vaiheessa on olemassa hyvin monia erityyppisiä e-oppimismenetelmiä, jotka toki oppilaitoksesta riippuen voivat olla enemmän tai vähemmän käytössä. Tämä on kuitenkin ala, joka tällä hetkellä on kovassa nousussa ja kehitysmahdollisuuksiakin on runsain mitoin. On siis järkevää harkita osan oppimateriaalin siirtämistä elektronisesti käsiteltäväksi, ja tätä hyödynnetään esimerkiksi luvussa 4 esiteltävissä MATLABin alkeissa.

E-oppijat on toisinaan luokiteltu 2000-luvulla digitaaliajan oppijoiksi perustuen heidän tuntemukseensa elektroniikan hyödyntämisestä ja digitaaliajan erikoisuuksista. Näillä e-oppijoilla on yleensä esimerkiksi seuraavia ominaisuuksia: sujuvuus monenlaisen eri median käytössä, kokemukseen perustuva aktiivinen oppiminen, useiden eri digitaalisten tietolähteiden yhtäaikainen käyttö ja käsittely, sosiaalisten tietoliikenneverkostojen ja sovellusten käyttö, epälineaarinen tietolähteiden käyttö tiedonhaussa, lähes kokoaikainen internet-yhteys sekä tiedonjakamiskulttuuri. [18] Edellä mainittujen perinteisempien ja vanhanaikaisempien oppimismenetelmien rinnalle on siis aiheellistakin nykyään olla olemassa uudentyyppistä oppimateriaalia, joka tukee mahdollisimman hyvin uudenlaisia oppijoita. Selkeitä yhtenevyyksiä onkin havaittavissa edellä kuvattujen oppimateriaali- ja oppijaluokittelujen välillä, kuten oppijan kyky käyttää useita erilaisia sähköisiä tietolähteitä yhtä aikaa sekä oppimateriaalien sähköistyminen ja verkostoituminen esimerkiksi wikeihin ja verkkokursseihin.

Kaiken kaikkiaan e-oppiminen avaa monipuolisia mahdollisuuksia. Oppimistapahtumat voivat sijaita myös perinteisen luokkahuoneen ulkopuolella ja uuden asian oppiminen tietyssä ajassa voi olla pelkästään oppijan itsensä vastuulla. Toki myös ulkopuolisen ohjaajan läsnäolo sekä e-oppiminen jossain rajatussa tilassa ovat mahdollisia, jos niille näkee käyttöä. Nämä mahdollisuudet ovat uudenlaisten oppijoiden oppimisen paranemisen kannalta olennaisia havaintoja, koska tätä kautta opettamisessakin voidaan kehittyä uudenlaiset haasteet kohdatessa. Toiminnallisesti hyvä e-oppimateriaali on teknisesti helppokäyttöistä ja ulkoasultaan pedagogisia ja sisällöllisiä tavoitteita tukeva [1, s. 11].

2.2 E-oppimisen taustalla oleva pedagogiikka

Pedagogiselta kannalta e-oppimateriaaleja voi arvioida sen mukaan, onko oppiminen tiedonhankintaa, osallistumista vai tiedonluomista. [1, s. 10] Tiedonhankinta tässä tapauksessa kuvastaa perinteistä oppimista, joskin tällä kertaa e-oppimateriaalin avustuksella, osallistuminen esimerkiksi koululehden tekoa eli osallistumista toimittamisen kulttuuriin ja tiedonluominen voisi tarkoittaa esimerkiksi uusien Wikipedian artikkelien kirjoittamista. [1, s. 10] E-oppimateriaalien taustalla olevaa pedagogiikkaa voisi kuitenkin määritellä edellä esiteltyä tarkemmin, koska äsken luetellut periaatteet eivät vielä ota huomioon e-oppimisessa juuri muuta kuin vain oppimateriaalin opiskelemisen tai tuottamisen. Yksikään e-oppimateriaali tuskin pystyy noudattamaan täydellisesti kaikkia hyväksi havaittuja pedagogisia periaatteita, mutta eri osasten yhdisteleminen on kuitenkin mahdollista.

Pedagogisia teorioita e-oppimisen taustalla voi ajatella olevan moniakin erilaisia, mutta monet näkökulmat e-oppimisen pedagogiikasta voi tiivistää ainakin kolmeen seuraavista: behaviorismi, kognitivismi ja konstruktivismi. Behavioristiset eli yksilötasoisen käyttäytymisen huomioon ottavat näkökulmat korostuvat ennen kaikkea siksi että iso osa e-oppimisesta voidaan tehdä itsenäisesti opiskeltavaksi, jolloin luonnollisesti myös yksilötason oppimisen hyväksi tehtävät asiat tulisi hyvän e-oppimateriaalin tapauksessa ottaa huomioon. Kognitivistiset käsitteet liittyvät osin behaviorismiin pyrkimyksessään selittää käyttäytymismalleja. Ne menevät kuitenkin vielä vähän pidemmälle analyysissään, koska kognitivismin tavoitteena on antaa selitys sille, mitä yksilön ajatustasolla tapahtuu tiettyjä toimintoja tehdessä ja miten oppimiseen liittyvät asiat voidaan selittää mielen toimintojen mukaisesti. Tämäkin teoriasuuntaus on e-oppimisen kohdalla ymmärrettävissä, sillä erityisesti itsenäisesti opiskeltavan materiaalin suunnittelussa oppimisessa tapahtuvat ajatuksenjuoksut on mahdollisesti otettava hyvinkin tarkasti huomioon. Konstruktivististen ideaalien mukaan puolestaan ihmiset muodostavat tietoa ja merkityksiä oppimilleen asioille sen mukaan, mitä he kokevat ja mitä ideoita heillä aiheeseen liittyen on entuudestaan. Tämäkin ilmiö tulee itsenäiseen opiskeluun tarkoitetussa e-oppimateriaalissa ehkä vahvimmin ilmi, koska opettamisen ohjaaja ei mahdollisesti ole ollenkaan läsnä, jolloin oppijalle itselleen jää enemmän tilaa muodostaa oma tietonsa. Kuten aiemmin mainittiin, nämä teoriat ovat yksi mahdollinen joukko, mihin e-oppimisen taustaan liittyvän pedagogiikan voisi luokitella eivätkä suinkaan välttämättä ainut mahdollisuus.

E-oppimisen merkityksen kasvu opetuksessa on myös synnyttänyt uusia pedagogisia teorioita vanhojen näkemysten rinnalle. Yksi tällainen on nimeltään konnektivismi, jonka idea pohjautuu siihen, että oppiminen tapahtuu osana sosiaalista verkostoa. Konnektivismissa tieto on jakautunut verkostoihin, ja oppiminen sekä tiedonkeruu tapahtuvat edellä mainittujen sosiaalisten verkostojen avulla, jotka voivat olla esimerkiksi nykyään suosiossa oleva sosiaalinen media. Konnektivismissa on tärkeää osata erottaa olennainen asia epäolennaisesta. Konnektivismi nojaa myös vahvasti siihen, että pääteksenteko itsessään on oppimisprosessi ja että tieto on luonteeltaan helpostikin muuttuvaa. [18, s. 20-21] Kokonaan uudentyyppiset oppimateriaalit voivat hyvinkin aiheuttaa tarpeen uusille oppimisen teorioille, joita esimerkiksi konnektivismikin edustaa.

2.2.1 E-oppimateriaali ja erilaiset pedagogiset mallit

Tarkastellaan seuraavaksi muutamia pedagogisia malleja ja niiden hyödyntämismahdollisuuksia e-oppimateriaalien kanssa. Nämäkin ovat edellisten lukujen tapaan vain ideoita, ja täysin oikeaa vastausta oikean tyyppiselle oppimiselle ei ole.

Tutkiva oppiminen

Tutkivan oppimisen ideana on opettajalla saada oppijat käsittelemään uutta aihetta, kun heidän lähtökohtinaan ovat omat ennakkokäsitykset ja aiempi tieto aiheesta. Tutkivan oppimisen työskentely etenee kysymysten asettamisella ja tietolähteistä hankittavan tiedon avulla yhteisöllisesti. [1, s. 93-94] Teknologian tukeman yhteisöllisen ja tutkivan oppimisen organisointia kuvataan pedagogisen infrastruktuurin mallilla, jonka keskeisimmät oppimisympäristön osa-alueet ovat seuraavat: tekniset rakenteet eli teknologian käytön järjestäminen ja tukeminen, sosiaaliset rakenteet eli yhteistyön käytännöt, tietoon liittyvät rakenteet eli tavat käyttää ja tuottaa tietoa sekä kognitiiviset rakenteet eli oppilaiden itsesäätelyn ja metakognitiivisten taitojen edistämisen tavat. [1, s. 95], [19, s. 113-114] Tutkivan oppimisen kannalta käytetyssä teknologiassa on tärkeää, että työkalut tukevat monipuolisesti ja joustavasti yhteisöllistä tiedonrakentelua eli käytännössä edistävät juurikin tutkivan oppimisen periaatteita [1, s. 96].

Keksivä oppiminen

Nimensä mukaisesti keksivän oppimisen päämääränä on saada oppija kokemaan keksineensä jotain henkilökohtaisesti itselleen uutta. Tätä kautta oppijalle annetaan mahdollisuudet kokea enemmän onnistumisen elämyksiä oppimisprosessinsa aikana opetettavan asian oppimisen lisäksi. [1, s. 100] Käytännössä tämä voi tarkoittaa yksinkertaisesti sitä, että vaikkapa tietotekniikka-aiheisessa tehtävässä annetaan tehtävänanto koodattavalle ohjelmalle sekä jotain tehtävänannon aiheeseen liittyvää oheismateriaalia ja sen jälkeen oppijalle vapaat kädet koodata ohjelma näiden pohjatietojen avulla mieleisekseen. On hyvin monia erilaisia tapoja, joilla keksivää oppimista voi tukea e-oppimateriaaleilla. Tärkeintä on kuitenkin mahdollistaa, että oppijat voivat kokeilla ja tutkia aihealuetta konkreettisesti ja lisäksi on olennaista tarjota monipuolisesti tietoa, jotta tutkittavaa aihetta voi tarkastella monesta eri näkökulmasta [1, s. 100].

Ongelmakeskeinen oppiminen

Ongelmakeskeinen, ongelmalähtöinen tai ongelmaperustainen oppiminen tarkoittaa pedagogista mallia, jonka mukaan oppijat perehtyvät opetettavaan aiheeseen huolellisesti pohtien laadittuja aihepiiriin liittyviä ongelmia. [1, s. 106] E-oppimateriaalilla tätä pedagogista mallia voi tukea esimerkiksi erillisellä prosessityökalulla, joka sisältää ongelmanratkaisun vaiheet, tai vaikkapa kokonaisella verkkokurssilla, jossa koko ongelmalähtöisen oppimisprosessin voi suorittaa alusta loppuun. [1, s. 109] Tällä mallilla työmenetelmät ovat melko vapaat e-oppimateriaalia hyödynnettäessä.

2.2.2 E-oppimisen opetusmetodien tarkastelu

Tarkastellaan opetuksellisesta näkökulmasta muutamia e-oppimiseen tarkoitettuja oppimateriaaleja, joita myöhemmissä luvuissa esiteltävissä kurssitoteutuksissakin on käytetty. Melko monissa toteutuksissa on käytetty hyväksi flipped classroom -opetusmenetelmää, jonka voi suomentaa käänteisopetukseksi. Tämän menetelmän perimmäisenä ideana on, että oppijat opiskelevat oppitunnin teorian itsenäisesti ja oppituntien aikana oppijat tekevät harjoitteita juuri opiskelemaansa aiheeseen liittyen. Opettajan rooli näillä oppitunneilla on ensisijaisesti ohjaava eikä autoritäärinen. Käänteisopetuksen oppimateriaaliin tulisi tässä tapauksessa panostaa riittävästi, koska yksilötason oppimista ei välttämättä tapahdu kaikille oppijoille samalla materiaalilla. E-oppiminen tarjoaa kuitenkin käyttökelpoisia mahdollisuuksia käänteisopetukseen. [18, s. 107] Käsitellään seuraavaksi tarkemmin muutama eri opetusmetodi.

Verkko-oppimisympäristö

Kurssin verkko-oppimisympäristöä suunniteltaessa ja rakennettaessa on hyvä päättää, miten verkkoympäristöä hyödynnetään [2, s. 48]. Verkko-oppimisympäristö soveltuu tiedonjakeluun, vuorovaikutukseen osallistujien välillä tai esimerkiksi opintosuoritusten tekoon ja sitä kautta opiskelijoiden arviointiinkin. Tällaiset pelisäännöt on hyvä miettiä entuudestaan verkkoympäristön kanssa, koska liiallinen määrä sisältöä ei välttämättä palvele tarkoitustaan. Osa opetusmateriaalista voi olla esimerkiksi järkevää jakaa muunlaisiin opetusmuotoihinkin, kuten perinteiseen kontaktiopetukseen. Hyvän oppimisympäristön valinnassa ja rakentamisessa pyritään ottamaan huomioon erilaiset vuorovaikutusmuodot [2, s. 48]. Vuorovaikutuksia tulisi olla sekä oppijoiden kanssa keskenään että oppijoiden ja opettajien välillä. Hyvät vuorovaikutustaidot mahdollistavat oppimisympäristön onnistumisen. Yksi erittäin olennainen piirre onnistuneessa verkko-oppimisympäristössä on myös liikkuvuuden helppous ja selkeys. Jämptillä ja selkeällä ulkoasulla sekä materiaalin loogispohjaisella sijoittelulla voivat verkkokurssin opiskelijat saavuttaa onnistuneita käyttäjäkokemuksia. Myöskin käytettävyydeltään heikko ympäristö vie aikaa paremmilta käyttökohteilta. [2, s. 49]

Screencast-videot

Screencast-videot eroavat normaaleista videoista siten että ne ovat ikään kuin videokuvakaappauksia tietokoneen ruudusta. Screencast-videoilla voi näppärästi havainnollistaa vaikkapa jonkin ohjelmiston toimintaa, sillä viedokuva rajautuu koko ruudun alueelle vähintään helpommin kuin erillisellä videokameralla kuvattuna. Screencast-videoiden teko ei vaadi useimmiten ilmaisohjelmistoja ihmeellisempiä sovelluksia. Näitä videoita voi esimerkiksi edellä esitellyssä käänteisopetuksen menetelmässä käyttää paljonkin hyödyksi. Videon kautta voi saada kattavamman multimediaoppitunnin oppijoiden koteihin, jos esimerkiksi kuvaa luentokalvoja ja opettajan puhetta yhtäaikaisesti. [18, s. 104-107]

Multimedia-aineistot oppimateriaaleina

Oppimateriaalina voi käyttää paljon erilaista sähköistä multimediamateriaalia, kuten esimerkiksi luentokalvoja tai tieteellisiä artikkeleja sähköisessä muodossa. Periaatteessa multimediaoppimateriaaliksi voi lukea kaiken materiaalin, mitä esimerkiksi omaan verkko-oppimisympäristöönsä lataa. Materiaalin määrässä on syytä erityisesti huomioida visuaalinen selkeys, helppokäyttöisyys ja opetuksellinen toimivuus. Pelkästään iso määrä erityyppistä oppimateriaalia ei itseisarvollisesti välttämättä ole oppijalle hyödyllistä, jos sen tarkoitusta ei ole miettinyt tarkemmin. [2, s. 51-52]

3. E-OPPIMISEN PERIAATTEIDEN HYÖDYNTÄMINEN LASKENTAOHJELMISTOJEN OPETUKSESSA

Tässä luvussa tarkastellaan, miten luvussa 2 esiteltyjä opetuksellisia periaatteita on hyödynnetty eri laskentaohjelmistojen opetussisältöihin eri oppilaitoksissa. Pääpaino on kuitenkin MATLAB-oppimateriaalin käsittelyllä. Tämä johtuu siitä että MATLAB on tämän työn pääasiallinen laskentaohjelmisto mutta myös siitä että MATLAB on nykyään yleisin korkeakouluissa käytössä oleva teknillisen laskennan työkalu.

3.1 R-ohjelmointikieli

R-nimistä ohjelmointikieltä käytetään tilastolaskennassa ja graafisten ympäristöjen suunnittelussa. Tilastolaskennan puolelta R pitää sisällään muun muassa lineaariset mallit, epälineaariset regressiomallit ja klusteroinnin. Tähän kieleen on myös saatavilla muitakin lisäosia, jotka tuovat enemmän lisäominaisuuksia kielen käyttöön. R on jatkokehitelmä samantyyppisesti nimetystä S-nimisestä ohjelmointikielestä. [26] MATLABin tapaan R on komentorivipohjainen laskentatyökalu, jossa voidaan operoida myös vektoreilla, matriiseilla ja taulukoilla. R on itsessään myös oma ohjelmointikielensä. [27, s. 1-18] R on ikään kuin kevyehkö ilmaisversio MATLABista. Moni toiminto R-kielessä on hyvin samankaltainen kuin nykyään laajemmassa käytössä olevassa MATLABissa, ja R-kielellä on myös joitakin alkeiskursseja, joiden kautta sen käyttöä voi oppia. R-kieleen liittyen löytyy monia avoimia verkkokursseja, joissa opetellaan joko R:n käyttöä pelkiltään tai tilastollisessa analyysissä. Useimmat kursseista, jotka helpolla etsinnällä löytää, ovat rakenteellisesti hyvin samanlaisia. Yksi esimerkki näistä on Johns Hopkinsin yliopiston avoin verkkokurssi.

Johns Hopkinsin yliopistolla on avoimena verkkokurssitoteutuksena R-ohjelmoinnin kurssi, jossa opetellaan ohjelmoimaan R-kielellä sekä lisäksi käyttämään R-kieltä tehokkaaseen datan käsittelyyn. Kurssi on mitoitettu neljä viikkoa kestäväksi kokonaisuudeksi siten että jokaisena viikkona opiskelua on seitsemän ja yhdeksän tunnin välillä. Neljälle opetusviikolle opetettava asia on jaettu suunnilleen tasaisesti niin, että ensimmäisellä viikolla lähdetään liikkeelle odotetusti R:n yleiskuvasta ja erityyppisistä datamuuttujista, joista jatketaan funktioiden, tietokantojen ja silmukallisten funktioiden kautta aina datasimulointiin saakka. Kurssin aiheista on opetusvideoita, pienimuotoisia tenttejä sekä laajamittaisempia ohjelmointitehtäviä. Kurssin suorittamisesta saa erillisen todistusasiakirjan. Etäopetuksen lisäksi kurssilla on myös keskustelupalsta, jossa opiskelijat voivat jakaa kokemuksia ja ongelmakohtia keskenään. [28]

Tämänmuotoinen kurssitoteutus on melko tavanomainen laskentaohjelmistojen opetuksessa, kuten myöhemmissäkin luvuissa huomataan. Verkko-oppiympäristö on tietomääränsä puolesta järkevästi suunniteltu, ja vuorovaikutussuhteet sekä opiskelijoiden että kurssihenkilökunnan välisesti löytyvät. Lisäksi laajemmat ohjelmointitehtävät tukevat vähintään opiskelijoiden keksivää oppimista, koska niissä on mahdollisuudet kokeilla monia erilaisia ratkaisumalleja. Kaiken kaikkiaan Johns Hopkinsin yliopiston pienimuotoinen verkkokurssi on hyvin perusvarma tapa opettaa R-laskentaohjelmiston käyttöä.

3.2 Octave

GNU Octave on korkeatasoinen ohjelmointikieli, jonka pääasiallinen käyttötarkoitus on numeerinen laskenta. Octaven syntaksi ja kieli ovat hyvin samankaltaisia kuin MATLABissa ja Octavea käytetäänkin monesti ilmaisena korvikkeena MATLABille. Monet Octavella kirjoitetut ohjelmat on samanlaisten komentojen vuoksi mahdollista siirtää suoraan MATLABiinkin. Octavea käytetään R:n tapaan komentoriviltä käsin. Octaven ilmaisuuden vuoksi sen luojat kannustavat käyttäjiä koodaamaan lisäominaisuuksia tähän laskentaohjelmistoon, ja jonkin verran niitä onkin tehty. [29] Octave soveltuukin hyvin MATLABin kaltaisen ulkoasunsa vuoksi ilmaiseksi korvaavaksi laskentaohjelmistoksi MATLABille. Octavesta puuttuu monia toimintoja MATLABiin verrattuna, mutta monet teknillisen laskennan perusteista voi hyvin harjoitella esimerkiksi Octaven avulla. Octaven käyttöönkin on olemassa oppimateriaalia, jota seuraavaksi käydään hieman läpi.

3.2.1 MIT:n numeerisen laskennan avoin kurssi

MIT (Massachusetts Institute of Technology) ei ainakaan avoimessa verkkokurssitarjonnassaan tarjoa opiskeltavaksi nimenomaan Octaven käyttöön liittyvää kurssia mutta oppilaitoksen numeerisen laskennan kurssilla kannustetaan laskentaohjelmana käyttämään nimenomaan Octavea tai MATLABia. Kurssin tiedoissa tosin mainitaan myös, että kurssin suorittaminen ei ole täysin sidonnainen mihinkään erityiseen ohjelmistoon, mutta matriisilaskenta-algoritmien hyödyntämisen vuoksi edellä mainitut ohjelmistot ovat suositeltavia. [15] Erityisesti Octaven käyttöä varten MIT:n kurssiopettaja on tehnyt kolme opetusvideota, joissa harjoitellaan Octaven perustoiminnallisuuksia. Videot ovat tyypiltään screencast-videoita ja ne etenevät loogisesti niin, että ensimmäisessä videossa käsitellään perusteita, kuten muuttujien alustusta ja siitä jatketaan muun muassa kuvaajien piirtämiseen ja käsittelemiseen sekä datasovitteiden tekoon. Videoiden sisällöt myös tukevat kurssin sisältöjä. [16]

Tätä kurssia ei ole erityisesti rakennettu Octaven ympärille, vaan Octave on pikemminkin vain suositeltu työkalu kurssin avuksi. Pääpaino kurssilla on tieteellinen laskenta, jossa ohjelmistoa voi käyttää apuna. Osa kurssimateriaalista on kuitenkin suunniteltu Octaven tukemisen kannalta hyvin niin että kurssin opiskelijalla on mahdollisuus oppia Octaven käyttöäkin.

3.2.2 GNU Octaven ohjemateriaali

Tämä ei ole opiskeltava verkkokurssi vaan GNU-käyttöjärjestelmän tarjoama kattava dokumentaatio Octavesta. Dokumentaatiossa esitellään monia eri Octaven toimintoja ja komentoja pelkästään raakatekstimuodossa. Raakatekstin sekaan on upotettu Octaven komentoja hieman erinäköisellä fontilla, jotta ne erottuvat muun tekstin seasta. [30] Dokumentaatio on toki ensisijaisesti tarkoitettu ohjemateriaaliksi eikä niinkään välttämättä oppimateriaaliksi. Sivustoa voi kuitenkin käyttää myös oppimateriaalina, koska se tarjoaa paljon esimerkkejä, joilla Octaven käyttöä voi opetella alkeista lähtien. Jos dokumentaatiota tarkastelee oppimateriaalina, niin siinä on erityisesti huomioitu multimedia-aineiston selkeys. Aineisto ei yritäkään edes olla millään tasolla visuaalisen yliampuva, vaan se täyttää tarkoituksensa siinä mitä se pyrkiikin olemaan. Sivustolla on myös erityisesti huomioitu kohderyhmä. Octave on Unix-pohjainen laskentaohjelmisto, ja monet Octaven käyttäjistä ovat todennäköisesti enemmän tai vähemmän tottuneet Unix-pohjaisiin käyttöjärjestelmiin sekä sitä kautta myös jossain määrin pelkistettyihin materiaaleihin. Monet näistä käyttäjistä eivät välttämättä edes tarvitse mutkikkaammin toteutettua oppimateriaalia, vaan heille riittää tällainen yksinkertaisen näköinen oppimateriaali, kuten tässäkin dokumentaatiossa on esitetty.

3.3 MATLAB

MATLAB on nykyään maailmanlaajuisesti miljoonien insinöörien ja tiedemiesten käyttämä teknillisen laskennan, mallintamisen ja visualisoinnin sekä muun muassa signaalin- ja kuvankäsittelyn laskentaohjelmisto. [3] MATLAB on laskentaohjelmistona onnistunut vakiinnuttamaan asemansa useissa yliopistoissa ja korkeakouluissa pääasiallisesti käytettävänä laskentaohjelmistona, vaikka MATLAB-lisenssi onkin maksullinen toisin kuin edellä esitellyt ohjelmistot. MATLABin suosion voivat selittää oikeanlainen markkinointi MathWorksin osalta, ohjelmointikielen yksinkertaisuus ja MATLABin toimintojen monipuolisuus. Totuus kuitenkin on, että MATLABilla pystyy tekemään paljon esimerkiksi juurikin teknillisen alan mallinnuksia, mistä syystä sen käyttötaitojen opetteleminen on jatkoa ajatellen hyödyllistä teknillisen alan opiskelijoille. Käsitellään edellisiä kappaleita tarkemmin joitakin MATLABiin kiinteästi liittyviä oppimateriaaleja ja kurssitoteutuksia.

3.3.1 TTY:llä ennestään oleva MATLAB-opetus

MATLABin alkeiden opetus on TTY:llä ennen toteutettu osana perusopintojaksojen eli insinöörimatematiikkojen ja laajojen matematiikkojen sisältöjä. Varsinainen oppimateriaali alkoi toisesta peruskurssista ja jatkui kaikissa ensimmäisen opiskeluvuoden peruskursseissa. Jokaisessa näistä kursseista oli tietty määrä PC-harjoituksia, jotka käytännössä olivat MATLAB-harjoituksia. Myös muita laskentaohjelmistoja, kuten Maplea, saatettiin tilanteesta riippuen käyttää, mutta MATLABia käytettiin eniten. Näihin harjoituksiin oli tietyt viikottaiset ajat, jolloin niitä sai käydä tekemässä. Harjoitukset pidettiin TTY:n tietokoneluokissa erillisen tuntiassistentin johdolla.

Harjoitusten sisältö liittyi kurssin muuhun oppimateriaaliin hyvin kiinteästi. Enimmäkseen näitä harjoituksia käytettiin havainnollistamaan peruskurssin matematiikkaa laskentaohjelmiston avulla. MATLABin alkeiden opetteleminen tapahtui siinä sivussa. Tehtävät myös suurimmaksi osaksi koostuivat kopioitavista komennoista, eikä erikseen alkeiden opettelemiselle jäänyt paljoa sijaa. MATLABin perustoimintojen opettelu jatkui kurssilla nimeltä *Johdatus teknilliseen laskentaan MATLABilla*, jossa ensimmäinen viikkoharjoitus käsitteli peruskomentojen kertausta. Opintojakson tarkoitus ei kuitenkaan ollut opettaa nimenomaan MATLABin käyttöä, vaan sen ensisijainen oppimistavoite oli teknillisen laskennan oppiminen, ja MATLABia käytettiin kurssilla teknillisen laskennan ohjelmistona. Kurssi ei myöskään ollut kaikille pakollinen, mutta sen pystyi sisällyttämään osaksi matematiikan syventäviä opintoja.

MATLABin perusteiden opetteluun ei juurikaan palattu enää myöhemmillä matematiikan tai signaalinkäsittelyn kursseilla. Niissä oletettiin, että monet esimerkiksi tämän diplomityön kuvaaman alkeiskurssin asioista osattiin jo entuudestaan, vaikka niiden opetteluun ei paljoa käytetty aikaa muuhun kurssin oppimäärään verrattuna.

Erillisistä PC-harjoituksista luovuttiin olosuhteiden pakosta. TTY:llä ei ennen lukuvuotta 2013-2014 ollut merkittävän paljoa juuri MATLABin alkeiden opetteluun keskittyvää oppimateriaalia, vaan iso osa siitä jäi opiskelijan omalle oppimisvastuulle muiden kurssien ohessa.

3.3.2 Tarton teknillisen yliopiston MATLAB-kurssi

Tartton yliopisto tarjoaa MATLABin johdantokurssia MOOCina (Massive Online Open Course). Kurssimateriaali sijaitsee Moodle-sivulla, jonne kuka tahansa voi rekisteröityä. Suoritus on mahdollista saada loppukevään aikana, ja siihen vaaditaan harjoitusten ja tentin tekemistä. Itseopiskelumateriaaliksi tarjotaan ulkopuolisia lähteitä eli tässä tapauksessa Mathworksin dokumentaatiota ja luentoina on videoluennot. Videoluennot ovat screencast-videoita MATLABin käytöstä, ja niissä puhuja puhuu samanaikaisesti kuin kirjoittaa MATLABilla. Samantyyppisiä opetusvideoita löytää esimerkiksi Mathworksin sivuilta ja Youtubesta. [4] Harjoituksia on viisi kappaletta, ja suoritus vaatii niiden kaikkien ja tentin tekemistä. Kurssisuoritus vaatii lisäksi, että jokaisesta harjoituksesta ja tentistä saa tietyn määrän pisteitä. Harjoitukset ovat palautettavia tehtäviä, ja palautukset voivat esimerkiksi olla MATLAB-skriptiä. [10]

Kaiken kaikkiaan kurssin Moodle-sivu muodostuu kuudesta alakohdasta ja jokainen alakohta on mitoitettu yhdelle viikolle opiskeltavaksi. Viidelle ensimmäiselle viikolle on kaksi luentovideota ja harjoitukset videoiden aihealueista, kun taas kuudennelle viikolle on ainoastaan yksi luentovideo ja harjoitusten sijaan on lopputentti. Kurssin suorittamalla saa Tarton yliopiston todistuksen. [10]

Ensimmäisen viikon materiaali lähtee alkeista liikkeelle. Tavoitteena todennäköisesti on saattaa MATLAB tutuksi sellaisillekin opiskelijoille, jotka eivät laskentaohjelmistoa ole ikinä ennen käyttäneet. Oppimateriaali jatkuu näistä alkeista aina pienin askelin eteenpäin. Luentovideoissa muun muassa opetettava asia etenee johdonmukaisesti, ja aikaisempien videoiden materiaalia, eli esimerkiksi jo luotuja funktioita, käytetään kätevästi hyväksi uuden asian opettamisessa. Kaikki videot ovat yli kymmenen minuuttia pitkiä, ja asia käydään hyvin perinpohjaisesti läpi useiden esimerkkien avulla. Videoiden puhuja myös tekee toisinaan virheitä videoissa eikä niitä ole leikattu pois lopullisista versioista. Luentojen ja harjoitusten sisältö on tarkemmin eritelty taulukossa 3.1.

Taulukko 3.1 *Tarton yliopiston MATLABin johdantokurssin sisältö viikottain. [10]*

Viikko	*Videoluennot*	*Luentojen asiasisältö*	*Harjoitukset*
Vk 1	Luento 1	MATLABin ikkunat, skalaarimuuttujien alustus, peruslaskutoimitukset, puolipisteen käyttö, clc	Sanallisia kysymyksiä, esimerkiksi puolipisteen merkitys
	Luento 2	Vektori- ja matriisimuotoiset muuttujat laskutoimituksineen, matriisien alkionhakutoiminto	
Vk 2	Luento 3	MATLABin valmiita funktioita, zeros, ones, rand, eye sekä funktion koodaamista ja ajamista, .m-tiedostot	2×2-matriisien peruslaskutoimituksia
	Luento 4	input, disp, num2str ja niiden soveltaminen funktioissa, if-lauseet loogisine operaattoreineen, funktio toisen funktion sisällä	
Vk 3	Luento 5	for-silmukka, ohjelman kommentointi, lukujonojen muodostaminen kaksoispistenotaatiolla	4×4-matriisi, josta MATLAB-funktiolla selvitetään rangi sekä lasketaan matriisiyhtälö
	Luento 6	while-silmukka, syvennetään funktio-osaamista mutkikkaammilla skriptoilla	
Vk 4	Luento 7	Lisää loogisia operaattoreita, kuten and, or ja not sekä niiden hyödyntämistä funktioiden ehtolauseissa	Kirjoitetaan funktio, joka antaa syötteenä annetusta vektorista parilliset ja parittomat alkiot erillisinä vektoreina
	Luento 8	Polynomit, polynomin juurten laskenta, lisäksi mm. komennot poly, polyval, polyder ja polyint, symbolinen laskenta (syms)	
Vk 5	Luento 9	Lisää funktioita ja symbolista laskentaa, derivointia, integrointia, jacobin matriisi	Mallikuvaajia, jotka on tarkoitus piirtää MATLABilla samanlaisina
	Luento 10	Kuvaajien piirtäminen plot-komennolla, kuvaajien käsittely sekä komento- että figureikkunasta	
Vk 6	Luento 11	Soveltavia esimerkkejä tilastotieteestä, tilastollisia tunnuslukuja, kuvaajien lukemista MATLABilla, helpin käyttö	Tentti

Tarton yliopiston materiaali on perinteinen tapa opettaa uuden laskentaohjelmiston käyttöä. Luennot ovat ainoastaan videomuodossa, joten kurssin opiskelijat ehdollistetaan katsomaan kaikki ja siinä sivussa mahdollisesti vähän sisäistämäänkin uutta asiaa. Palautettavat skriptit mahdollistavat osaltaan opiskelijoiden keksivän oppimisen kokemuksia.

3.3.3 MIT:n MATLAB-kurssi

MIT tarjoaa avoimena kurssimateriaalinaan MATLABin johdantokurssia. Kurssin on tarkoitus tarjota opiskelijoille sujuvat MATLAB-taidot niin ohjelman käytössä kuin myös käytetyimmissä työkalupakeissa. Kurssi on mahdollista suorittaa aina tammikuisin, ja aikaa suoritukseen kuluu neljä viikkoa. [11] MIT:n kurssi ei opeta MATLABin perusteita muuten kuin tarjoamalla johdannon eri komentojen syntakseihin ja perustoimintoihin ulkopuolisilla pdf-tiedostoilla [20], [21]. Luentomuistiinpanot ja harjoitustehtävät on saatavilla pdf-muotoisina myös kurssin luennointiajankohdan ulkopuolella. Taulukossa 3.2 on karkeasti eritelty jokaisen luennon ja harjoituksen sisältö.

Taulukko 3.2 *MIT:n MATLABin johdantokurssin sisältö. [12]*

Luennot	*Luentojen asiasisältö*	*Harjoitukset*	*Harjoitusten asiasisältö*
Luento 1	Johdanto, MATLAB-skriptat, muuttujien alustus, muuttujien laskutoimitukset, kuvaajien piirtämisen perusteet	Harjoitus 1	Kahdeksan lyhyttä tehtävää skalaari- ja vektorimuuttujista sekä kuvaajien piirtämisestä, neljä vapaaehtoista pitkää skriptatehtävää
Luento 2	Funktiot, loogiset operaattorit, if, for, while, kuvaajien käsittelyä, kuvaajien syventämistä	Harjoitus 2	Seitsemän tehtävää, joissa luodaan omia pienimuotoisia funktioita ja piirretään kuvaajia datan visualisoimiseksi sekä neljä vapaaehtoista pidempää skriptatehtävää
Luento 3	Lineaarialgebraa, matriisiyhtälöitä, polynomeja, polynomisovitteiden tekemistä, optimointi, derivointi ja integrointi, differentiaaliyhtälöitä	Harjoitus 3	Viisi aiempaa selkeästi pidempää ja soveltavampaa tehtävää, joista erityisesti viides on monivaiheinen prosessi sekä lisäksi yksi pitkähkö vapaaehtoinen tehtävä
Luento 4	Tilastoanalyysi, tietotyypit, kuvatiedostojen luku, skriptojen debuggaus	Harjoitus 4	Kahdeksan luennon aiheisiin liittyvää tehtävää, yksi vapaaehtoinen tehtävä, joka jatkaa edelliskerran vapaaehtoisesta tehtävästä
Luento 5	Johdantoa symbolisen laskentaan, Simulinkiin, tiedostojen lukemiseen ja käsittelyyn ja GUI:hin		

MIT:n opetusmateriaali on erittäin pitkälle vietyä ja haasteellista verrattuna moniin muihin MATLABin johdantokurssien oppimateriaaleihin. Nämä johdannot tarjoavat jo hyvin kattavan osaamisen MATLAB-ohjelmiston käytössä, ja perusteiden opettelemiseen ei käytetä paljoa aikaa ennen kuin niiden soveltamista haastavampiin esimerkkeihin käydään läpi. Joka luennon alussa myös kysytään opiskelijoilta mietteitä edelliskerran tehtävistä ja niiden haastavuustasosta. Tämä onkin melko tavallinen käytäntö Yhdysvaltojen kalleimmissa yliopistoissa ja korkeakouluissa, joissa opiskelijan opinnoissa edistymisestä pidetään paikoitellen paljonkin huolta. Lisäksi myös edellisen harjoituskerran tavallisimpiin virheisiin puututaan ja käydään niitä läpi. Kurssin suoritusvaatimuksissa toisaalta myös mainitaan, että opiskelijoilla tulisi olla perusymmärrys ohjelmoinnista, lineaarialgebrasta, differentiaaliyhtälöistä ja todennäköisyyslaskennasta. [13, s. 3] Luentokalvojen rakenteesta käy ilmi, että opiskelijoiden oletetaan osaavan erityisesti ohjelmoinnin peruskäsitteitä entuudestaan, jolloin heidän opiskelija-aineksensa tilanne on erilainen meihin nähden, missä MATLABia opetetaan ennen kuin useimmille välttämättä on ohjelmointia opetettu ollenkaan.

Luennot ovat melko laajoja. Luentokalvoista ilmenee, että niissä käytetään varsinaisen teoriaopetuksen lisäksi myös paljon havainnollistavia kuvia, jotka voivat olla niin kuvakaappauksia MATLABin ikkunoista kuin myös itse piirrettyjä jotain asiaa havainnollistavia kuvia. Luennot myös etenevät melko kovalla vauhdilla, eikä yhteen asiaan kerrallaan käytetä paljoa aikaa. Luennoilla opetetusta asiasta käydään läpi myös melko paljon esimerkkitehtäviä ja harjoituksia. Vaikka monessa asiassa mennään jopa melko pitkälle teoriassa, niin silti monessa kohtaa opetettavat asiat myös väännetään rautalangastakin, kuten siinä vaiheessa kun puhutaan polynomien vektoriesityksistä [14, s. 10]. Siinä vaiheessa luentoja on jo takana sen verran asiaa, että opiskelijat saattaisivat hyvin tajuta kyseisen asian itsekin. Yksi merkittävä eroavaisuus muihin alkeismateriaaleihin nähden kurssisisällössä on se, että opetus alkaa MATLAB-skriptojen kirjoittamisesta ja jatkuu vasta sen jälkeen muuttujien alustuksella ja muulla yleensä tutummalla alkupään oppimateriaalilla. Aritmeettiset laskutoimituksetkin tulevat vasta ensimmäisen luennon loppupuolella. Kurssin kannalta olennainen asia käydään läpi neljässä ensimmäisessä luentokalvokokonaisuudessa. Näiden lisäksi on myös ekstrakurrikulaarista materiaalia varten viides luentokalvokokonaisuus.

Harjoitukset ovat hyvin vahvasti soveltavia ja melko pitkäkestoisiakin tehtäviä. Niissä nojataan paljon siihen, että opiskelija osaa hakea tarvittavista komennoista omapäisesti tietoa joko MATLABin helpin tai luentokalvojen avulla. Luentokalvoissakin kerrottu teoria on melko vähäistä eikä se välttämättä riitä paljon soveltamista vaativien tehtävien ratkontaan. Ensimmäisissä harjoituksissa opiskelijaa vielä ohjataan tehtävänannossa käyttämään lähinnä valmiita tiettyjä komentoja toteutukseen, mutta myöhemmissä harjoituksissa opiskelijalle annetaan jo vapaat kädet ratkaista tehtäviä riittävällä toiminnallisella tasolla. Vapaaehtoisiksi merkityt tehtävät ovat jo vaatimustasoltaan melko pitkälle vietyjä ottaen huomioon, että kyseessä on johdantokurssi. Ensimmäisten harjoitusten kuusi ensimmäistä tehtävää ovat tavallisia erimuotoisten muuttujien alustus -ja laskutehtäviä, joiden ratkaisemiseen ei MATLABiin hiukan vihkiytyneeltä todennäköisesti kulu kauaa aikaa. Seitsemännessä tehtävässä harjoitellaan MATLABin valmiita funktioita vektoreille ja matriiseille, ja kahdeksannessa tehtävässä harjoitellaan kuvaajien piirtämistä ja käsittelyä. Neljä viimeistä tehtävää ensimmäisissä harjoituksissa ovat muutamaan otteeseen mainittuja vapaaehtoisia tehtäviä, joiden päämääränä on soveltaa opittuja perusteita ja funktioita. Tehtävät ovat monivaiheisia, ja jokaisessa vaiheessa kerrotaan, millä tavoin seuraava ominaisuus toteutetaan. Tehtävät vaativat paljon asiaan perehtymistä eivätkä ne välttämättä ole täysin yksinkertaisia MATLABiin enemmän perehtyneille.

Toisissa harjoituksissa sallitaan jo enemmän vapaavalintaisuutta tehtävien ratkaisuissa. Ensimmäisissä harjoituksissa opiskelijan täytyi vielä käyttää tehtävänantojen mukaisia komentoja, mutta toisissa harjoituksissa on jo sallittua käyttää mitä tahansa tapaa ratkaista tehtävä, kunhan ratkaisu toimii tehtävänannon mukaisesti. Tehtävänannoissa kuitenkin on myös apufunktioita, joita ratkaisuun voi käyttää. Kaikissa tehtävissä ei anneta ollenkaan apuja, vaan opiskelijan odotetaan itse ottavan asioista selvää. Osaan tehtävistä ei myöskään suoraan löydy luentomateriaalista ratkaisua, vaan opiskelijan odotetaan osaavan soveltaa paljon jo opittuja menetelmiä. Jotkin tehtävät myös vaikuttavat olevan useapamanakin vuotena kierrätettyjä, koska esimerkiksi MathWorksin keskustelupalstoilla on vuonna 2011 kysytty neuvoa hyvin samantyyppiseen tehtävään kuin mitä yksi harjoitustehtävä on [5].

3.3.4 MathWorksin MATLAB-johdantomateriaali

MATLAB-ohjelmiston laatinut MathWorks-yritys tarjoaa myös paljon erityyppistä johdantomateriaalia itse kehittämälleen ohjelmistolle. MathWorks on laatinut oman interaktiivisen aputyökalun nimeltä MATLAB Academy. Tämä alusta tarjoaa MATLAB-lisenssin omistajille interaktiivisia kursseja, joiden sisältönä on erinäisiä ohjelmademoja, harjoituksia ja tenttejäkin. Tentit on rakennettu niin, että käyttäjä voi itse kirjoittaa MATLABin tyyliseen ohjelmaan vastauksensa, jonka ohjelma arvostelee. Kursseja voi käydä siinä tahdissa kuin itselle parhaiten sopii, ja materiaali on eritasoista. Tällä materiaalilla voi harjoitella niin perusteita kuin edistyneempiäkin toimintoja. Perusteissa harjoitellaan alkeita, kuten syntaksia sekä tiedostojen tuomista, viemistä ja analysointia. Edistyneemmässä materiaalissa on mukana jo koodin kirjoittamistakin. [6]

MATLAB Fundamentals on kolmipäiväinen maksullinen MATLAB-käyttötaitojen opettelukurssi. Kurssikuvauksessa sanotaan, että sinne voi osallistua, vaikkei olisi yhtään aiempaa kokemusta MATLABista tai edes ohjelmoinnista. Kurssilla käsitellääan muun muassa MATLABin käyttöliittymää, skalaari- vektori- ja matriisimuotoisia muuttujia, datan visualisointia, ulkopuolisten tiedostojen datan analysointia sekä skriptojen ja funktioiden kirjoittamista. Kurssilla on opetusmateriaalina käytössä opetusvideoita oppimateriaaliin perehtymiseen ja myös melko ainutlaatuinen työkalu muihin tässä työssä tarkasteltuihin oppimateriaaleihin verrattuna. Fundamentals pitää sisällään interaktiivisen MATLABin komentoikkunan, johon voi vastauksena kirjoittaa komentoja, jotka kone tulkitsee oikeiksi tai vääriksi. Näiden osioiden suorittamiseen ei tarvita omaa MATLAB-lisenssiä. Tämä työkalu on hyvä opettamaan MATLAB-syntaksia erityisesti sellaisille, joilla ei mitään aikaisempaa taustaa laskentaohjelmistoista ole. Tämän avulla opiskelija pääsee suoraan näkemään oman vastauksensa oikeellisuuden sen sijaan että hän joutuisi etsimään tekemänsä virheen muualta, mikä on erityisesti aloitteluvaiheessa hyvä ominaisuus. Fundamentals kuitenkin on noin tuhansissa euroissa liikkuvan hintansa suhteen melko kallis kurssi ainakin yksityiskäyttäjälle, joten sille ei välttämättä runsasta käyttöä nähdä. [7]

MATLAB Academyn ja Fundamentalsin lisäksi MathWorks tarjoaa myös perinteistäkin oheismateriaalia. Sivuilla on tarjolla havainnollistavaa dokumentaatiota, jonka avulla ennen MATLABiin koskematonkin voi tutustua ohjelmistoon. Materiaali lähtee liikkeelle aloitusnäkymästä ja jatkuu syntaksin opettamisen kautta aina funktioihin, kuvaajiin ja pienimuotoiseen ohjelmointiinkin. [8] Tekstipohjaisen dokumentaation lisäksi sivustolla on kattava arkisto opetusvideoita MATLABin käyttöön liittyen [9]. Opetusvideoiden lisäksi tarjolla on joitakin toisenlaisiakin videoita, joiden perimmäinen tarkoitus ei ole juuri opettaa. Nämä ovat esimerkiksi joitakin kilpailuaiheisia tuotoksia. Suurin osa videoista on MathWorksin itsensä laatimia. Myös keskustelupalsta ohjelmistoaiheisille kysymyksille löytyy. Pääasiassa se toimii vertaistuen kautta, eli toiset käyttäjät voivat itse vastata muiden käyttäjien esittämiin kysymyksiin.

MathWorks tarjoaa hyvin laajan kattauksen erilaista opetusmateriaalia niin ohjelman johdantoon kuin myös vaativampiinkin toimenpiteisiin liittyen. Apumateriaali on kasattu sivustolle ehkä hieman epäjohdonmukaisesti, sillä esimerkiksi videot löytää eri paikasta kuin muut asiat, vaikka videotkin ovat suureksi osaksi nimenomaan opetuspainotteisia. Tällä materiaalilla pääsee kuitenkin erittäin hyvin jyvälle MATLABin moniulotteisesta toiminnasta, jos on aikaa itsenäiselle opiskelulle ja harjoittelulle.

3.3.5 Muita MATLABin alkeismateriaaleja

Useissa teknillistä laskentaa opetustarjonnassaan tarjoilevissa korkeakouluissa on myös jonkinlainen kokonaisuus MATLABin alkeiden opetukselle. Kaikki nämä kokonaisuudet eivät kuitenkaan ole avoimia muille kuin kyseisen korkeakoulun opiskelijoille, joten tässä luvussa mainitaan vain muutamia verkkokursseja ja- materiaaleja, jotka ovat löydettävissä hakukonetulosten joukosta.

Nashville Tennesseen Vanderbilt University tarjoaa monenlaisia MOOCeja eri aihealueista, ja moni näistä alueista liittyy myös ohjelmointiin. Vanderbilt Universityn tarjoama MATLABin johdantokurssi on noin yhdeksän viikkoa kestävä MOOC, jonka kesto on keskimäärin 4-6 tuntia viikossa. Kurssin tavoitteena on ensisijaisesti opettaa tietokoneohjelmointia sellaisille, joilla ei entuudestaan aiheesta paljoa kokemusta ole, ja opetusohjelmistona käytetään MATLABia. Kurssin alustuksessa perustellaan MATLABin käyttöä ohjelmoinnin alkeiden opetukseen sillä, että se on helppo oppia, monipuolinen ja käyttökelpoinen insinööreille. Kurssin päätavoitteena ei siis niinkään ole oppia MATLABin käyttöä, mutta materiaali kuitenkin edesauttaa myös MATLABin perustoimintojen oppimista. Lisäksi sisältö ei ole matemaattisesti haastavaa. Matemaattiselta pohjalta riittää ymmärrys esikorkeakoulutason matematiikasta. Myös kiinnostus ohjelmointia kohtaan olisi hyvä olla. Kurssin sisältö keskittyy MATLABin osa-alueissa muun muassa erityyppisiin muuttujiin, funktioihin ja silmukoihin, jotka ovat peruspalikoita melkein missä tahansa muissakin ohjelmointikielissä. [22]

Toinen esimerkki on jo iäkäs dokumentaatio, jonka Utahin yliopiston David Eyre on kasannut [23]. Tämä vuodelta 1998 peräisin oleva verkkosivu käsittelee muutamia MATLABin perustavanlaatuisia ominaisuuksia ja komentoja. Tietohan ei luonnollisesti ole merkittävästi vanhentunut vuosien saatossa, mutta MATLABin komentonäkymä on todennäköisesti kokenut muutoksia. Tämä sivu kuitenkin keskittyy ensisijaisesti komentojen opetteluun, ja se tarjoaakin komentojen käytön harjoittelua varten kätevästi kopioitavia ja liitettäviä tekstipätkiä, jotka MATLABin komentoikkunassa toteuttavat toimintoja. Välissä on myös sanallisesti kerrottu, mitä mikäkin komento saa aikaiseksi MATLABissa. Sivu käsittelee muun muassa erityyppisiä muuttujia ja niiden välisiä laskutoimituksia, funktioita, kuvaajien piirtämistä, datatiedostojen käsittelyä sekä ohjelmoinnin perusteita. [23] Vaikka nykyaikana ehkä mieluummin käytettäisiin joitakin modernimpia opetusvälineitä, niin loppujen lopuksi myös yksinkertaisilla välineillä, kuten tällä sivulla, voi ihan yhtä hyvin opettaa tietyn tyyppisille opiskelijoille samat alkeet. Tämä sivu osoittaa lähinnä sen, että ei MATLABin opetettava aines liiemmin vanhene, vaikka ajat ja ohjelmistot hieman muuttuisivatkin.

4. MATLABIN ALKEIDEN OPETUKSEN KOKONAISUUDEN KEHITTÄMINEN TTY:LLÄ

4.1 Taustaa

TTY:llä tarjottua MATLABin alkeiden opetusta päätettiin uudistaa luvussa 3.3.1 mainituista syistä. Tavoitteena oli luoda kokonaisuus, josta pystyi ennen kaikkea oppimaan MATLABin perustoiminnot mutta myös kätevästi testaamaan toimintojen osaamista opiskelijoilla siten että opetushenkilökunnan aikaa säästyisi. Lisäksi kätevä ominaisuus uudistetulle MATLABin alkeiden opetukselle olisi myös keino palata myöhemmässä vaiheessa opintojaan kertaamaan vanhoja asioita, jos niitä mahdollisesti tarvitsee silloin käyttää. Myös spesifimpää MATLAB-opetusta uudistettiin pilotin kautta.

MATLABin alkeiden opetuksen uudistaminen oli osa TTY:n MATLABin opetuksen kehittämisen pilottihanketta, johon osallistuivat TTY:n laitoksista ne, jotka MATLABia huomattavissa määrin opetuksessaan käyttävät. Nämä ovat matematiikan laitos, fysiikan laitos, signaalinkäsittelyn laitos, automaatio- ja systeemitekniikan laitos sekä TTY:n Porin sivuyksikkö. Pilottihanketta suunniteltiin kevään 2015 aikana ja matematiikan laitoksen tehtäväksi tuli tässäkin opinnäytetyössä tarkemmin kuvattu ensimmäisen vuoden opiskelijoille pakollinen MATLABin alkeiden uudelleen suunnittelu. Muut pilottiin osallistuneet laitokset laativat omaan opetustarjontaansa soveltuvaa MATLAB-oppimateriaalia. MATLABin alkeiden osalta pilottia kokeiltiin ensimmäistä kertaa jo syksyllä 2015 kaikille TTY:n uusille opiskelijoille.

TTY:n muut laitokset tekivät MATLAB-pilottiin omaan oppiainekseensa sopivaa uutta materiaalia. Signaalinkäsittelyn laitos teki pilottiin screencast-videoita, joissa oli sisäänrakennettu tehtävänanto. Videoilla käytiin läpi signaalinkäsittelyn kannalta hyödyllistä MATLAB-osaamista ja samalla videolla esitellään myös aihealueeseen liittyvä tehtävä. Videoilla on mahdollisesti alustusta tehtävään eli sitä tehdään jonkin verran eteenpäin jo videolla itsessään tai sitten muu tarvittava osa materiaalista on saatavilla muissa materiaaleissa kuin videoilla. Käytännössä opiskelijoiden on kuitenkin katsottava video kokonaisuudessaan läpi, jotta he pääsisivät tehtävässä alkuun. Fysiikan laitoksen materiaalit keskittyvät luonnollisesti fysiikan ilmiöihin. Fysiikan laitoksen oppimateriaalit sisältävät pdf-tiedostoja, joissa on hieman sekaisin perinteisiä fysiikan laskutehtäviä ja MATLABilla ratkaistavia tehtäviä. MATLAB-tehtävissä annetaan esimerkiksi valmiiksi jokin m-tiedosto, jonne täydennetään tehtävänannon mukaisesti vaikkapa fysikaalisten suureiden laskukaavoja. Tehtävissä saatetaan myös hyödyntää erilaisissa tiedostomuodoissa olevaa mittausdataa, jota pitää MATLABilla käsitellä. Systeemitekniikan laitos käytti omassa MATLABin johdantokokonaisuudessaan palautettavaa komentoketjua eli skriptiä, johon opiskelijan pitää täydentää tehtävänannon mukaisesti oikeita tietoja. Sen lisäksi on Moodlen automaattisesti tarkastettavia tehtäviä, joissa testataan opiskelijan ymmärrystä MATLABin komennoista. Palautetut skriptit tarkastaa opettaja. Nämä muutkin laitokset saattavat opetuksessaan hyödyntää omien materiaaliensa lisäksi MATLABin alkeiden kurssia.

4.2 Osaamistavoitteet ja ydinaines

Luvussa 4.1 mainittujen tavoitteiden pohjalta TTY:n MATLABin alkeiden opetus toteutettiin erillisenä Moodle-sivuna, jonne uusi oppimateriaali kerättiin. Moodlesivusta tehtiin kaikille opiskelijoille avoin eikä sinne tarvittu erillistä kurssiavainta rekisteröitymiseen. Tämä mahdollisti hyvin alkeiden kertaamisen myöhemmässä vaiheessa opintoja, sillä Moodle-sivulle voi vapaasti liittyä. Itse suoritusosuus toteutettiin automaattisesti tarkastettavien STACK-tehtävien avulla. STACK-tehtävien toiminnasta kerrotaan lisää luvussa 4.3. Itsestään tarkastavat ja arvostelevat STACK-tehtävät mahdollistivat vähäisen tarpeen opetushenkilökunnalle, koska tehtävien tarkastamiseen ei tarvittu juurikaan ihmisnäkökulmaa. MATLABin käyttötaitojen testaaminen oli siis helppoa noin 700:n opiskelijan suoritusten tarkastamisessa.

Alkeista pyrittiin luomaan kokonaisuus, josta olisi keskimäärin kaikkein eniten hyötyä opiskelijoille sekä MATLAB-taitojen opettelussa että myöhemmässä vaiheessa opintoja. Lisäksi tavoitteena oli luoda luvun 2 mukaisesti pedagogisesti järkevä kokonaisuus. Vaikutteita otettiin paljon luvussa 3 esitellyistä laskentaohjelmistojen oppimateriaaleista. Suunnitteluvaiheessa yleishyödyllisimpinä ja perustelluimpina oppimisosa-alueina pidettiin seuraavia MATLABin toimintoja:

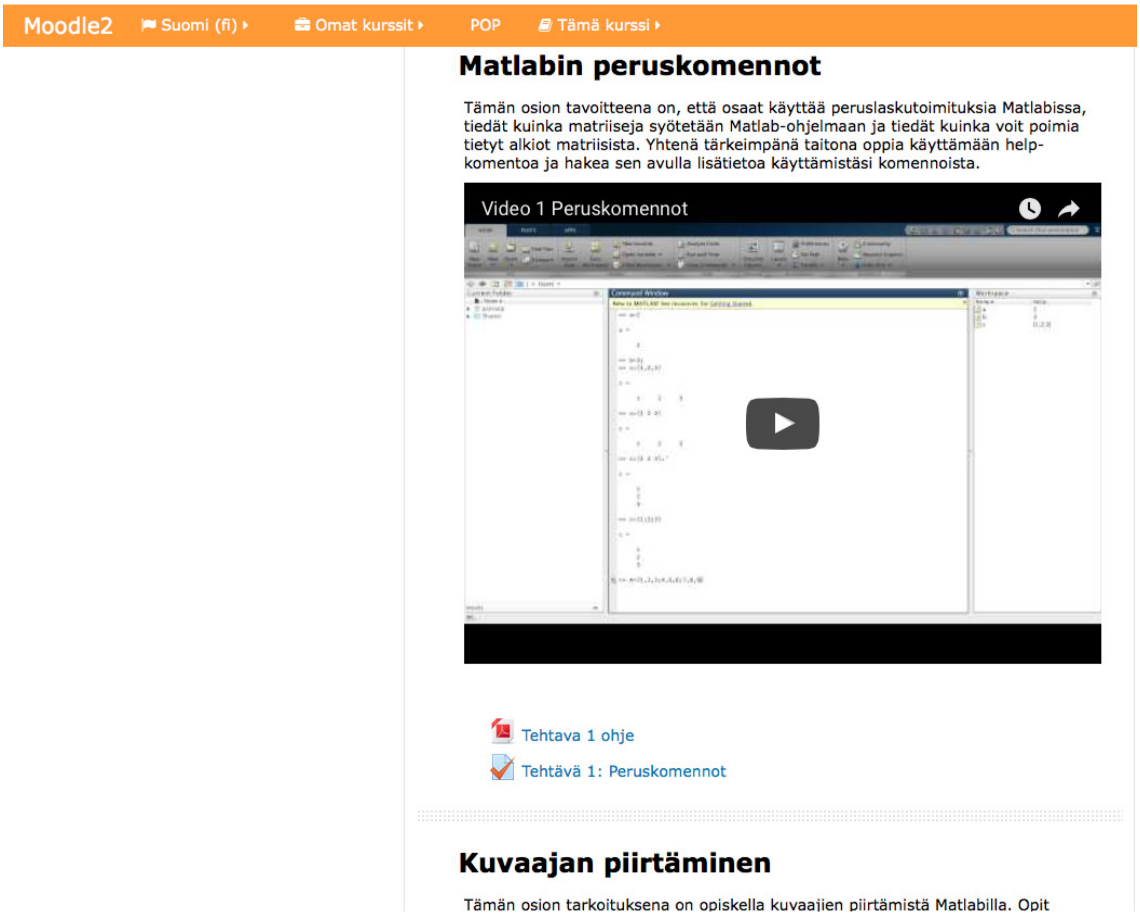
1. MATLABin erityyppiset muuttujat, kuten skalaarit ja matriisit, peruslaskukomennot ja help-toiminnon käyttö
2. Kuvaajien piirtäminen ja niiden käsittely
3. Editori-ikkunan käyttö, omien komentoketjujen ja funktioiden ohjelmoiminen

4. MATLABin sisäänrakennettujen funktioiden käyttö, sovitteet ja pienimmän neliösumman menetelmä

5. Tiedostojen lukeminen ja käsittely MATLABissa, data import

Jokaisesta ylläolevasta osa-alueesta luotiin Moodle-toteutukseen oma osionsa, johon laadittiin sekä arvosteltava STACK-tehtävä että oheismateriaalia, jossa käsiteltiin aihealuetta laajemmin kuin tehtävän ratkaisemiseen vaadittiin. Oheismateriaali oli aiheeseen liittyvät screencast-videot, joihin oli nauhoitettu MATLABin käyttöä, sekä ohjepdf-tiedostot. Laaditut ohjepdf-tiedostot löytyvät liitteistä 6-10. Tavoitteena oli luoda jokaisesta viidestä tehtävästä kokonaisuus, jonka pystyi ohjemateriaalin sekä muun matematiikan kurssimateriaalin avulla itsenäisesti ratkaisemaan. Myös ohjausta tarjottiin MATLAB-harjoitusten muodossa, mutta TTY:n opiskelijoilleen tarjoaman ilmaisen MATLABin kampuslisenssin vuoksi pyrittiin luomaan kokonaisuus, jossa jokainen opiskelija pystyi tarvittaessa ratkaisemaan tehtävät itsekin. Yllä luetellut osa-alueet myös pyrittiin määrittämään niin, että niistä olisi mahdollisimman suurta hyötyä opiskelijalle hänen myöhemmissä opinnoissaan opiskelualasta riippumatta. Tehtävien sisällöllä koitettiin saada opiskelija tiedostamaan MATLABin kätevyys laskentaohjelmistona, jotta opiskelija myöhemässä vaiheessa opintojaan saattaisi myös opittujen alkeiden siivittämänä palata ohjelmiston pariin. Moodle-sivulla oli materiaalin lisäksi keskustelupalsta, jossa opiskelijat pystyivät neuvomaan toisiaan tai kysymään vastuuhenkilöiltä neuvoja tehtäviin.

Verkko-oppimisympäristöä suunniteltaessa otettiin huomioon luvussa 2 esiteltyjä piirteitä verkko-oppimisympäristöistä. Moodle ulkoasultaan tukee kokonaisuuden selkeyttä, koska jokainen yllä mainitusta viidestä osa-alueesta on jaettu omaan erilliseen aiheeseensa, josta löytyy kaikki tehtävään liittyvä oheismateriaali. Moodlesivun ulkoasua voi katsoa kuvassa 4.1. Vuorovaikutustaitoja parantava keskustelupalsta on sijoitettu erilleen muista aiheista sivun yläosioon. Screencast-videot ovat hyvä keino edistää jokaisen opiskelijan itsenäistä oppimista, mikä oli myös yksi pilotin tavoitteista. Automaattisesti tarkastettavat tehtävät edistävät opiskelijoiden keksivän oppimisen kokemuksia. Kaiken kaikkiaan koko tämä multimediamateriaali pyrittiin jäsentämään selkeästi, ja sitä ei ollut tarkoituskaan olla liikaa, koska liiallinen määrä oheismateriaalia menettää helposti olennaisen viestinsä. MATLABin alkeissa on käytetty luvun 3 mukaisesti samantyyppistä materiaalia, mitä monessa muussa kurssissa oli käytetty ja lisäksi käsiteltävä oppiaines on suurelta osin samaa kuin monissa luvussa 3 esitellyissä kokonaisuuksissa.



Kuva 4.1 *Osa lopullisen Moodle-sivun näkymää MATLABin alkeissa. Kuvassa näkyy Youtubeen upotettu opetusvideo sekä linkit ohjepdf-tiedostoon ja STACK-tehtävään.*

4.3 STACK-tehtävät

4.3.1 Taustaa

STACK tulee sanoista System for Teaching and Assessment using a Computer algebra Kernel. Tämän järjestelmän on luonut Edinburghin yliopiston professori Chris Sangwin. STACK-tehtäviä voi integroida osaksi Moodle-oppimisalustan tenttejä, mikä mahdollistaa tehtävien itsenäisen suorittamisen. Erikoisuutena STACK-tehtävissä tavallisiin Moodle-tehtäviin verrattuna on niiden kyky käyttää tietokonealgebraa (Computer Algebra System, lyh. CAS) vastausten oikeellisuuden tarkastamiseen. Esimerkiksi yksinkertaisen algebrallisen yhtäläisyyden STACK tarkistaa seuraavalla logiikalla: [17, s.9-18]

```
If
  simplify(sa-ta) = 0
then
  true
else
  false.
```

Ylläolevassa ehtolauseessa `sa` merkitsee opiskelijan anatamaa vastausta ja `ta` vastaavasti opettajan eli tehtävän laatijan ennalta määräämää oikeaa vastausta tehtävälle. Ylläoleva ehto palauttaa väitteen totena, jos `sa`:n ja `ta`:n välinen erotus on nolla, ja vastaavasti epätotena, jos erotus poikkeaa nollasta. Myös matemaattiset konstruktiot, kuten liukuluvut, matriisit ja polynomit ovat STACKin avulla mahdollisia. Tämä onkin selkeästi monipuolisempi valikoima itsetarkastavissa tehtävissä kuin Moodlen oletuksena tarjoamat monivalintatehtävät. [17, s.9-18]

Toinen STACKin vahvuuksista on älykäs satunnaistus. Satunnaistuksen vuoksi tehtävistä voi luoda jokaiselle opiskelijalle uniikkeja. Tällä ehkäistään sitä että opiskelijat katsoisivat vastauksensa toisiltaan. Myöhemmin tässä työssä ilmoitetut satunnaistustehtävät eivät ole toteutettu täsmälleen samoin oikeissa opiskelijoilla testatuissa tehtävissä. CAS ja STACK-järjestelmän pohjana olevan Maxima-ohjelmiston funktiot mahdollistavat STACK-tehtävissä käytetyn satunnaistuksen. Satunnaistusfunktiot ovat seuraavat Maximassa käytetyt funktiot:

```
rand(n): generoi kokonaisluvun väliltä 0...n-1
rand(n.0): generoi liukuluvun väliltä 0...n
rand([a,b,...,z]): valitsee satunnaisen luvun listasta
rand(matrix(..)): asettaa jokaisen matriisin alkion satunnaiseksi.
```

STACK-tehtävät antavat opiskelijan vastauksesta myös automaattisesti palautetta ja pisteyttävät tehtävät. Palautteen STACK-järjestelmä antaa tiettyjen vastauspuiden (potential response tree) kautta, jotka käyttäjä pystyy itse määrittelemään mieleisikseen. Nämä vastauspuut ovat eräänlaisia peräkkäisiä ehtolauseita, joiden avulla pystyy automaattisesti tarkastamaan erilaisia tilanteita, antamaan niistä oman palautteensa sekä pisteyttämään vastaukset eri tavoin riippuen opiskelijan antamista vastauksista. Jos opiskelija esimerkiksi antaa vastauksensa sellaisessa muodossa, josta tehtävänannon rakenteen perusteella pystyy päättelemään tietyllä tavalla virheellisen ajatuksen kulun, niin vastauspuun voi määritellä antamaan opiskelijalle palautteena vinkkejä, miten virheestä pääsisi eroon. Kuten yllä on tullut mainittuakin, STACK ymmärtää myös vastauspuissa tietokonealgebraa, joten vastauspuun opiskelijan vastauksia ja oikeita vastauksia voi muokata mieleisikseen matemaattisin operaatioin. [17, s.9-18]

Pisteytyksen voi myös määrittää erilaiseksi jokaiselle eri vastauspuulle. Pisteytyksessä mahdollistuvat kätevästi esimerkiksi eri pistemääräiset a)- b)- ja c)-kohdat tai osittaisten pisteiden antaminen osittain oikeasta vastauksesta. Vääriin vastauksiin voi myös halutessaan määrittää rangaistuksia (penalty), jos kokee tietyllä tavalla väärän vastauksen ansaitsevan pistemenetyksiä.

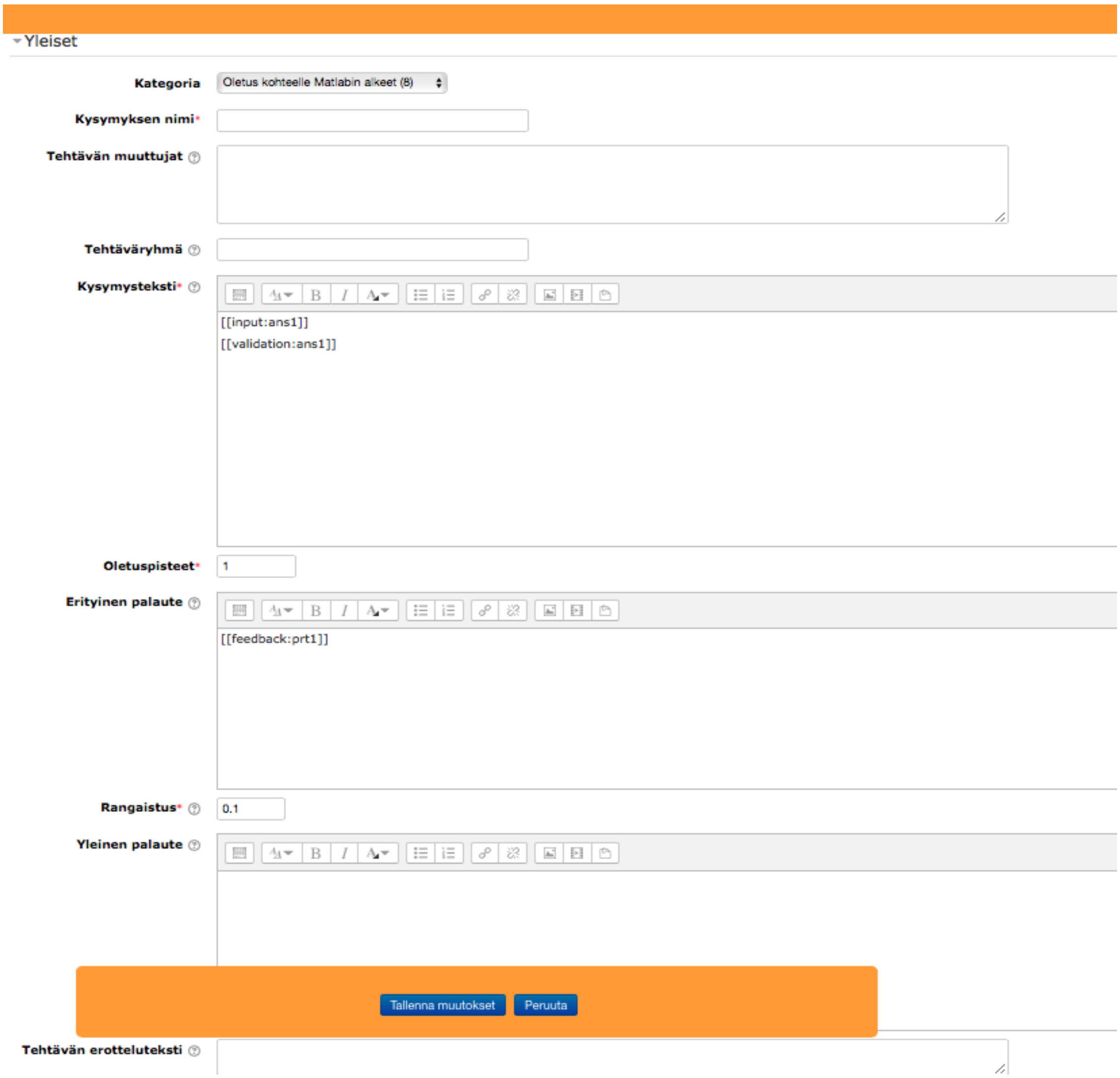
4.3.2 STACK-tehtävän luominen

STACK on mahdollista asentaa osaksi Moodlea. Kun tämän operaation on tehnyt, Moodleen voi luoda yhtenä tenttitehtävätyyppinä STACK-tehtävän. Tentin asetuksia pystyy muokkaamaan normaaliin tapaan Moodlessa eli esimerkiksi tentin kokonaispisteet voi määrittää sitä kautta ja myöskin tentissä voi olla useita STACK-tehtäviä. STACK-tehtävässä määritellyt pistemäärät ja moodle-tentin pistemäärät skaalautuvat näillä määrittelyillä keskenään yhteen. Erilaiset pisteytykset näissä kahdessa instanssissa voivat kuitenkin olla hieman hämääviä, joten voi olla järkevää pisteyttää sekä STACK-tehtävät että Moodle-tentit samoin periaattein.

STACK-tehtävä luodaan Moodlesta löytyvällä käyttöliittymällä. Käyttöliittymän yläosan voi nähdä kuvassa 4.2. Jokainen yksittäinen STACK-tehtävä tarvitsee siis käytännössä ohjelmoida toimimaan. Iso osa STACKin taustalla toimivasta matematiikasta on peräisin Maximasta ja MathJaxista, joten sama Maximassa käytetty syntaksi suurimmaksi osaksi toimii myös STACKissa. Joihinkin tekstikenttiin tehtävää luodessa tulee pelkästään raakatekstiä mutta tiettyihin tekstikenttiin tulee kirjoittaa Maximan mukaista koodiakin.

Kohdassa *tehtävän muuttujat* (question variables) alustetaan tehtävissä käytössä olevat muuttujat. Nämä muuttujat voivat olla esimerkiksi skalaareja, listoja tai matriiseja. Tässä kohdassa alustettuja muuttujia käytetään myöhemmässä vaiheessa tehtävää esimerkiksi määrittämään oikea vastaus. Tässä kohdassa myös määritetään mahdolliset satunnaistukset haluttuihin muuttujiin. Kysymystekstissä täällä alustettuihin muuttujiin voi viitata ympärillä olevilla @-merkeillä.

Kysymysteksti on STACK-tehtävässä se osio, jonne suurin osa käyttäjälle näkyvästä tekstistä kirjoitetaan. Kysymysteksti ymmärtää LaTeX-muotoiluja, joten matemaattisen tekstin kirjoittaminen STACK-tehtävään sujuu hyvin. Kysymystekstin sekaan upotetaan myös vastauslaatikot, joihin tehtävän tekijä antaa vastauksensa. Vastauslaatikoiden syntaksi on `[[input:ans1]]`, missä `ans1` on vastaussyötteen muuttuja. Saman tekstin tilalla voi luonnollisesti olla joku toinenkin muuttujan nimi. Jokaista vastaussyötettä kohden kysymystekstissä pitää olla myös vastauksen arviointi eli teksti `[[validation:ans1]]`.



Kuva 4.2 *STACK-tehtävän luomiseen tehty käyttöliittymä. Ulkoasu on vastaava kuin Moodlessa muutenkin.*

Vastauspuiden käyttöön ottoa varten pitää joko kysymystekstiin tai kohtaan *erityinen palaute* (specific feedback) kirjoittaa `[[feedback:prt1]]`, `[[feedback:prt2]]` jne. riippuen siitä kuinka monta eri vastausehtolausetta haluaa luoda. Suositeltavaa on alustaa vastauspuut erityiseen palautteeseen kysymystekstin sijaan. Yleisesti ottaen jos tehtävässä on vastausmuuttujia, jotka eivät liity toisiinsa mitenkään, niin näihin toisiinsa liittymättömiin vastauksiin vaaditaan oma vastauspuunsa. Rangaistuksen pistemäärän voi myös määrittää. Tämä ei suoraan tarkoita vielä, että rangaistus tulisi käyttöön vaan se kytketään myöhemmässä vaiheessa päälle, mikäli halutaan. *Tehtävän erottelutekstiin* (question note) on puolestaan pakko merkitä erikseen kaikki muuttujat, joissa käytetään satunnaistusta. Tällä kohdalla tehtävässä erottuvat eri kysymysversiot, joita muodostuu erilaisista satunnaisista tehtävänannoista.

Vastauksen tietoihin alustetaan, minkämuotoinen ja- tyyppinen mikäkin vastaus on. Edellä kerrottiin, miten vastaussyötteet lisätään vastaustekstiin. Jokaista vastaussyötettä kohti määritellään erikseen vastaustyyppi (algebrallinen vastaus, matriisi, oikein/väärin, tekstikenttä), mallivastaus, jonka voi ottaa esimerkiksi tehtävämuuttujista, sekä muutamia lisäominaisuuksia. Nämä lisäominaisuudet ovat esimerkiksi vastauskentän koko, joka kannattaa säätää sen mukaisesti, minkä kokoinen vastaus on, sekä desimaalien salliminen vastauksessa. Vastausosion täyttäminen STACK-tehtävän määrittelyissä määrittää, millaisena vastauskenttä näkyy lopullisessa tehtävässä ja millaisia vastauksia se hyväksyy.

Viimeinen tehtävän toimimisen kannalta olennainen asia on *vastauspuiden* määrittely. Jokaista vastauspuun solmua voi ikään kuin ajatella yhtenä ehtolauseena, joka antaa erilaiset vastaukset riippuen siitä, mitä käyttäjä vastaa. Jokaista erityisessä palauttessa näkyvää palautelohkoa kohden tulee näkyä myös vastauspuu tarvittavine tietoineen. Vastauspuuhun merkitään *käyttäjän antama vastaus* (student answer), joka otetaan vastauskentästä, *opettajan vastaus* (teacher answer), johon käyttäjän vastausta verrataan sekä *vastaustesti* (answer test), joka tarkoittaa sitä menetelmää, millä vastauksia verrataan toisiinsa ja todetaan vastauksen oikeellisuus. Näiden lisäksi vastauspuuhun merkitään oikeasta/osittain väärästä/väärästä vastauksesta saatavat pisteet sekä palaute kustakin vastauksesta. Tehtävän rakenteesta riippuen vastauspuun solmuja voi linkittää toisiinsa, jolloin voi muodostaa esimerkiksi moniosaisia tehtäviä. Vastaustestiä varten voi myös virherajoihin sallittavien desimaalien määrää säädellä.

4.3.3 Erilaisia STACK-tehtäviä

Tehtävien pääasiallinen tarkoitus on opettaa MATLABin peruskomentoja ja- toiminnallisuuksia luvussa 4.2 kerrottujen periaatteiden mukaisesti. Nämä tehtävät on suunniteltu ensisijaisesti TTY:n ensimmäisen vuosikurssin opiskelijoille osana pakollisia opintoja. Tehtävissä harjoitellaan esimerkiksi MATLABin peruslaskutoimituskomentoja, kuten skalaareilla ja vektoreilla kertomista, kuvaajien piirtämistä, funktioiden käyttöä ja excel-tiedoston datan tuomista ohjelmaan ja kyseisen datan hyödyntämistä. Tehtävät kokonaisuudessaan löytyvät liitteistä 1-5, mutta tässä kappaleessa esitellään jokaisen tehtävän taustoja erikseen.

Tehtävä 1: Peruslaskukomennot

Tämän tehtävän päämääränä on perehdyttää opiskelija MATLABin ulkoasuun ja tavanomaisimpiin komentoihin. Tehtävässä käydään läpi perustoiminnallisuuksia.

Hyvin suurella todennäköisyydellä näitä tehtäviä tekevä opiskelija ei ole koskaan ennen käyttänyt MATLABia, joten on hyvä lähteä liikkeelle siitä perusolettamuksesta, että mikään asia ei ole entuudestaan tuttua. Tämän tehtävän kannalta ei ole siis itsearvoista, että vielä pyritäänkään tekemään mitään erityisen mutkikasta, vaan ensisijainen tarkoitus niin tässä kuin tehtäväpankin neljässä muussakin tehtävässä on opettaa opiskelijalle MATLABin perustoimintamallit.

Ensimmäisen tehtävän oheismateriaalissa (enemmän tästä kappaleessa 4.4) lähdetään liikkeelle puhtaasti siitä, että opiskelijan ei oleteta tietävän MATLABin käytöstä mitään. MATLABin alkeiden suorittaminen ajoittuu toiselle perusmatematiikan kurssille, joissa yliopistotasolla käsitellään ensimmäistä kertaa matriisimuotoisia muuttujia ja niiden välisiä laskutoimituksia. MATLABin laskutoimitukset pohjautuvat matriisilaskentaan, joten on aiheellista käsitellä, miten matriiseilla lasketaan tällä laskentaohjelmistolla. Matriiseilla operoiminen ei toisaalta välttämättä heti aluksi ole intuitiivisella tasolla tuttua, mutta oheismateriaalilla pyritään hiukan selventämään yleisimpiä ongelmakohtia. Pitkin ensimmäistä tehtävää ja muitakin tehtäviä pyritään kannustamaan MATLABin `help`-toiminnon käyttöön, koska sen avulla pystyy helposti ohjelman sisällä jo pelkästään saamaan lisätietoa komentojen toimintatavoista.

Ensimmäinen tehtävä koostuu neljästä alakohdasta, joista kustakin saa 0,5 pistettä eli yhteensä 2 pistettä. Ensimmäisessä alakohdassa kysytään matriisituloa. Tulontekijöinä on 3×3-matriisi `A` sekä 3×1-vektori `b`. Matriisin `A` arvot ovat aina samat mutta kohtuullisen mielivaltaiset. Pienin alkio on 4 ja suurin 1337. Vektorin `b` alkiot on puolestaan satunnaistettu erilaisten tehtävänantojen saavuttamiseksi `rand`-funktiolla. Nämä alustetaan tehtävämuuttujissa siis seuraavasti:

```
A:matrix([4,5,6],[67,11,1000],[1337,101,77]);
b:[rand(20),rand(7),rand(111)];
```

Vastauslaatikkona on valmiiksi lopputuloksena saatava 3×1-vektori, johon opiskelijan tulee syöttää erikseen jokainen vastausvektorin alkio.

Toisessa alakohdassa testataan MATLABin alkioittaisten laskutoimitusten hallintaa. Pyrkimyksenä on saada ymmärtämään, mitä piste laskutoimituksen edessä tarkoittaa. Tehtävänantona on laskea edellä määritetyn matriisin `A` alkioittainen korotus toiseen potenssiin MATLABilla. Fontin tulostumisen vuoksi piste saattaa olla hieman vaikeasti havaittavissa laskutoimituksessa, mutta virheellisestä vastauksesta saatava palaute auttaa oikean vastauksen jäljille.

Kolmannen alakohdan on määrä testata matriisien alkion- ja rivinhakutoimintoja.

Tehtävässä kehotetaan opiskelijaa ensin kirjoittamaan MATLABiin esimerkiksi merkinnät `A(4)` ja `A(3,:)` ja sen jälkeen tulkitsemaan komentoikkunan tulosteista, mitä ne kirjoittamalla tapahtuu. Komennothan tulostavat matriisista `A` järjestyksessä neljännen alkion ja kolmannen vaakarivin vastaavasti. Yksittäisen alkion haku on satunnaistettu ylläolevan `rand`-funktion avulla siten, että matriisin `A` jälkeen tuleviin sulkuihin arpoutuu satunnainen kokonaisluku väliltä $[1, 9]$. Kun opiskelija on ideaalitilanteessa testannut tarpeeksi näitä hakutoimintoja, on aika tehdä itse tehtävä eli laskea MATLABilla seuraava laskutoimitus:

```
A(rand)*A(3,:)*b
```

Laskussa nähdään käytännössä sekä skalaarilla vektorin kertominen että vektorien kertominen toisillaan ja lisäksi myös ominaisuus käyttää laskutoimituksissa hakutoimintoja. Vastaukset ovat myös melko satunnaistettuja jokaisella opiskelijalla, koska myös vektorin `b` alkiot ovat alkujaan satunnaisia.

Neljännessä alakohdassa hyödynnetään matriisien välistä Kroneckerin tuloa matriisien dimensioiden hahmottamisessa. $m \times n$-matriisin A ja $p \times q$-matriisin B välinen Kroneckerin tulo $A \otimes B$ määritellään seuraavasti: [24]

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & & & \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

Tulomatriisista $A \otimes B$ muodostuu $(mp) \times (nq)$-kokoinen matriisi. Opiskelijalta vaaditaan tässä tehtävässä muodostamaan matriisi `C`, joka on pelkästään ykkösistä koostuvan 3×2-matriisin ja 10×1-pystyvektorin, joka sisältää järjestyksessä alkiot kokonaislukuväliltä $[1, 10]$, välinen Kroneckerin tulo. MATLAB-notaatiolla tämä toteutetaan seuraavasti:

```
C=kron(ones(3,2),(1:10)')
```

Sen jälkeen opiskelijaa pyydetään ilmoittamaan matriisin `C` korkeus (vaakarivien lukumäärä) ja leveys (pystyrivien lukumäärä). Tähän saa oikean vastauksen selville monella eri tavalla. Vastauksen voi päätellä Kroneckerin tulon määritelmästä, mikä toisaalta ei testaa niinkään MATLABin käyttötaitoja, mutta toisaalta tämä on vain yksi 0,5 pisteen alakohta, joten pieni oppiainerajat ylittävä sisältö ei ole vakavaa. Vastauksen voi myös katsoa suoraan vastausmatriisin `C` tulosteesta eli siitä voi laskea vaaka- ja pystyrivien lukumäärän. Toiset vaihtoehdot on katsoa matriisin koko MATLABin työtilasta (workspace) tai selvittää asia `size`-komennon avulla. Tämä alakohta soveltuu hyvin testaamaan eri MATLABin tapoja selvittää matriisien dimensioita. Käytetty laskutoimitus ei todennäköisesti ole entuudestaan tuttu, joten opiskelijat on hyvä päästää kokeilemaan toimintoja jollain tuntemattomalla operaatiolla. Ensimmäisen tehtävän voi kokonaisuudessaan nähdä liitteessä 1.

Tehtävä 2: Kuvaajien piirtäminen ja käsittely

Täman tehtävän kautta pyritään oppimaan piirtämään kuvaajia ja käsittelemään niiden ominaisuuksia MATLABin avulla. Kuvaajien käsittelyä hyvin todennäköisesti useimpien opiskelijoiden täytyy tehdä melkeinpä opintoalasta riippumatta, joten sen perusteita on hyvä hieman MATLABin kautta opettaa. MATLABin kuvaajanpiirto-ominaisuudet ovat oppimiskynnykseltään mahdollisesti hieman korkeat mutta kuitenkin huomattavasti monipuolisemmat moneen tavalliseen toimisto-ohjelmaan verrattuna.

Tehtävässä on tarkoitus onnistua piirtämään samanlainen kuvaaja, joka tehtävänannossakin näkyy. STACK-tehtäviin voi upottaa kuvaajia JSXGraph-kirjaston avulla. JSXGraph on muun muassa geometriaan, funktioiden kuvaajien piirtämiseen ja data-analyysiin keskittyvä JavaScript-pohjainen kirjasto selainkäyttöä varten. [25] JSXGraph pitää erikseen asentaa osaksi Moodlea, jotta STACK osaisi näyttää kuvat tarkoituksenmukaisesti. JSXGraph-kuvaajan pystyy tässä tapauksessa tekemään melkein täysin samanlaiseksi kuin MATLABillakin piirretyn. Piirrettiin funktion $f(x) = x\sin(x)$ kuvaaja siten että mittapisteitä oli 100 kappaletta ja pisteet olivat kokopunaisia timantteja. Kuvaajan arvot sijoitettiin välille $[0, 10]$. MATLABilla riittää siis luoda vektori `x`, joka sisältää arvot edelliseltä lukuväliltä 0,1:n välein. Kuvaajanpiirtokomentona riitti käyttää yksinkertaisesti `plot`-komentoa. STACK-tehtävässä annetaan suoraan kopioitavaksi seuraavat rivit:

```
x=0:0.1:10;
h=plot(x,x.*sin(x),'rd','MarkerFaceColor','r');
hold on;
grid on; grid minor;
get(h)
```

Piirretyn kuvaajan ominaisuudet tallennetaan MATLABin `handleen` nimeltä `h`, ja viimeisen rivin komento `get` hakee nämä ominaisuudet ja tulostaa ne komentoikkunaan. Opiskelijan on etsittävä, mitä lukee kohdassa `MarkerFaceColor` ja ilmoitettava se vastauksena tehtävään. Vastauslaatikkona on kolmen alkion mittainen vaakavektori, johon kirjoitetaan punaisen värin RGB-koodi. Puhtaan punaisen värin

RGB-arvo on [255,0,0], mutta MATLABissa nämä arvot on skaalattu välille $[0, 1]$, jolloin oikea vastaus on [1,0,0]. Tämä alakohta on tehty sillä idealla, että oikeaa vastausta ei saa selville kirjoittamatta vaadittuja komentoja MATLABiin. Vastauksen voi toisaalta tietotekniikkaan perehtynyt opiskelija kyetä päättelemäänkin, mutta se ei välttämättä ole edellisen tehtävän oppiainerajat ylittävän aineiston tavoin huono asia.

Toisessa alakohdassa kuvaajan akselit skaalataan uudelleen `axis`-komennolla, jossa on satunnaistetut arvot. Uudelleenskaalauksen jälkeen kuvassa näkyy vain osa edellistä kuvaajaa, ja opiskelijan tehtävänä on ilmoittaa kuvassa näkyvien punaisten pisteiden lukumäärä. Tehtävä on opiskelijan kannalta helppo ratkaista, mutta tehtävän tavoitteellisen toimimisen varmistamiseksi sen taustalla olevat matemaattiset operaatiot vaativat hieman säätämistä erityisesti satunnaistuksen vuoksi. Tehtävän muuttujat alustetaan seuraavasti STACK-notaatiota käyttäen:

```
xmin:5.05+rand(13)/10+rand(99)/10000;
pisteet:5+rand(10);
xmax:xmin+pisteet*1/10;
ymin:xmin*sin(xmin)-1;
ymax:xmax*sin(xmax)+1;
```

Yllä olevat muuttujat `xmin` ja `xmax` ovat `axis`-komennon uudelleenskaalauksen x-akselin minimi- ja maksimiarvot sekä `ymin` ja `ymax` ovat vastaavasti y-akselin minimi- ja maksimiarvot. Muuttuja pisteet on tehtävän oikea vastaus. Mittapisteet asettuvat kuvaajaan säännöllisin välein, tarkalleen ottaen 0,1:n välein, koska x-koordinaatin pisteet ovat tehtävässä 0,1:n välein välillä $[0, 10]$. Tätä tietoa hyödyntäen voidaan rakentaa x-koordinaatin ala- ja ylärajat `xmin` ja `xmax` siten että ne riippuvat satunnaistetusta muuttujasta `pisteet`. Tällä tavoin voidaan STACK-tehtävään tehdä lähes täysin satunnaisista arvoista koostuva tehtävänanto, joka kuitenkin osaa tulkita vastauksen oikein.

Toisen tehtävän kolmas alakohta ei hyödynnä paljoa STACKin ominaisuuksia. Tehtävässä täytyy palauttaa omalla opiskelijanumerollaan otsikoitu pdf-tiedosto ensimmäisen alakohdan mukaisesta kuvaajasta. Palautus tehdään erillisellä Moodlen palautustoiminnolla. Tehtävätekstissä annetaan seuraavat MATLAB-komennot:

```
op=xxxxxx; (x-kirjaimien tilalle tulee opiskelijanumero)
title(['Opiskelijanumeroni on ' num2str(op)])
print(gcf,'-dpdf','kuvatesti.pdf')
```

Sen jälkeen ainut STACKiin liittyvä osuus kohdassa on oikein/väärin -tyyppinen kysymys, jossa opiskelijan on vastattava rehellisesti, saiko hän palautettua omalla opiskelijanumerollaan varustetun pdf-tiedoston kuvaajasta. Vastaamalla `kyllä` saa pisteen ja vastaamalla `ei` ei saa pistettä. Tässä kohdassa myös painotetaan, että jos vilpistä jää kiinni, niin koko MATLABin alkeiden suoritus hylätään. Tämän alakohdan tarkoitus on lähinnä varmistua siitä että opiskelija todellisuudessa ratkaisee tehtävät itse eikä suoraan kopioi vastauksiaan ja sitä myötä myös väärää opiskelijanumeroa joltakulta toiselta. Toisen tehtävän ulkoasun voi kokonaisuudessaan nähdä liitteessä 2.

Tehtävä 3: Editori-ikkunan käyttö ja omien komentoketjujen (skriptien) kirjoittaminen

Tehtäväpaketin kolmannen tehtävän päämääränä on opettaa perusteita omien komentoketjujen ohjelmoimisesta ja ajamisesta MATLABilla. Lisäksi yksi olennainen osa tätä tehtävää on myös perehdyttää erillisen muokkausikkunan perusteet opiskelijoille. Myös ensimmäisessä tehtävässä esitellyn MATLABin päänäkymän aktiivinen kansio ja sen oleellisuus tulevat väistämättä vähintään yrityksen ja erehdyksen kautta tutuiksi. Tehtävän yhteydessä opiskelijan tulee ladata MATLABin p-tiedosto, joka on yksinkertainen funktio. Funktio on p-tiedosto sen takia, että opiskelija ei saisi selville, mitä se pitää sisällään. Kolmannen tehtävän ensimmäisenä alakohtana opiskelija ajaa MATLABilla tämän äsken ladatun tiedoston satunnaisella argumentilla ja ilmoittaa vastauksena MATLABin tulostaman luvun. Satunnaistuksella varmistetaan, että tehtävän vastaus on kaikille opiskelijoille erilainen, ja tiedostomuodon takia tehtävän ratkaisemiseen käytännössä vaaditaan MATLABin käyttöä. Funktiotiedosto sisältää seuraavat matemaattiset yhtälöt:

$$
\begin{aligned}
&l_{100} = max(\lfloor 10 \cdot rand \rfloor, 1) \\
&luku = \lfloor |\sin(arg1)| \rfloor \cdot 89 + 10 \\
&loppu = rand \\
&out = l_{100} \cdot 100 + luku + loppu,
\end{aligned}
$$

missä $rand$ tarkoittaa MATLABin satunnaisgeneroitua lukua, eli liukulukua reaalilukuväliltä $(0, 1)$, $arg1$ on funktion syöte, ja out on funktion paluuarvo. Sekä funktion syötteessä että paluuarvon laskutoimituksissa on satunnaistettuja arvoja. Paluuarvon laskuissa satunnaistus ei kuitenkaan vaikuta muihin kuin paluuarvon ensimmäiseen numeroon ja desimaaliosaan. Tämä huomataan siitä että luku $l_{100} \in [1, 10]$

kerrotaan luvulla 100 ja luku *loppu* on pelkästään MATLABin antama *rand*-syöte. Tällöin ainut tarkasteltava osa paluuarvoa on *luku*, joka riippuu ainoastaan funktion syötteestä $arg1$. STACK-tehtävässä täytyy siis vastaustestissä ottaa huomioon ainoastaan paluuarvon ykköset ja kymmenet. Tehtävän tulkitsemaa oikeaa vastausta $Tans$ pitää muokata siten, että se huomioi ainoastaan ykköset ja kymmenet vastauksesta:

$$Tans = \lfloor |sin(arg1)| \cdot 89 + 10 \rfloor - 100 \cdot \lfloor \frac{\lfloor |(sin(arg1)| \cdot 89 + 10 \rfloor}{100} \rfloor.$$

Opiskelijan vastauksen $Sans$ on myös vastattava täysin algebrallisesti edellä määritettyä oikeaa vastausta $Tans$. Seuraavassa yhtälössä ans on STACK-tehtävään annetun vastauksen arvo:

$$Sans = \lfloor ans \rfloor - 100 \cdot \lfloor \frac{ans}{100} \rfloor.$$

Tässä tilanteessa jos opiskelijan antama vastaus ans on sama kuin $luku$, niin vastaus on oikein. Periaatteessa siis riittää antaa oikeaksi vastaukseksi pelkästään ykköset ja kymmenet MATLABin antamasta lopputuloksesta, koska tehtävä tarkistaa ainoastaan niiden lukujen oikeellisuuden. Opiskelija kuitenkaan ei lähtökohtaisesti luultavasti ymmärrä tätä erikoisuutta ja vaikka ymmärtäisikin, niin se ei merkittävä puute tehtävän toiminnallisuuden kannalta olisi. Tehtävä kuitenkin osaa tulkita vastauksen oikeaksi myös MATLABin antamasta koko tuloksesta, mikä on olennaista.

Komentoketjujen harjoittelua varten on loogista, että tehtävässä pyydettäisiin opiskelijaa kirjoittamaan MATLABilla oma skripta. Tehtävän sisältö täytyi kuitenkin miettiä tarkkaan, koska STACK-tehtävien merkkijonojen tulkinta ei ole erityisen edistynyttä, joten vastauksena ei oikein pysty antamaan valmista MATLAB-koodia. Täytyi pyrkiä luomaan tehtävä, joka olisi sopivan haastava kaikentasoisille opiskelijoille ja jonka ratkaisemiseen tarvitaan editori-ikkunan käyttöä.

Loppujen lopuksi tehtävän rakenteeksi päätettiin, että opiskelijan on muodostettava funktio, joka osaa käyttäjän antamasta syötteestä määrittää diskriminantin avulla polynomin reaalisten juurten lukumäärän. Syöte on muodoltaan kolmialkioinen vektori, jonka alkiot kuvaavat toisen asteen polynomin termien kertoimia. Tehtävänannossa annetaan valmiiksi suurin osa valmista funktiota, johon on tarkoitus selvittää täytteet tyhjiin kohtiin.

Tehtävän ratkaistakseen opiskelijan on ensinnäkin ymmärrettävä diskriminantin yhteys toisen asteen polynomin juuriin. Toisekseen opiskelijan täytyy osata myös vähintään tulkita koodia oikein. Lisäksi ensimmäisen alkeiden tehtävän oppimateriaalista täytyy olla mielessä myös vektorimuotoisten muuttujien alkionhakutoiminto. Ensimmäisessä kohdassa täytyy palauttaa mieleen diskriminantin D laskukaava, joka menee seuraavasti:

$$D = b^2 - 4ac,$$

missä a, b ja c ovat toisen asteen polynomin p termien kertoimia eli vektorimuodossa ilmaistuna $p = \begin{bmatrix} a & b & c \end{bmatrix}$. Ensimmäiseksi funktioon on täydennettävä diskriminantin laskukaava eli käytännössä kertoimien a, b ja c paikat on määritettävä oikeiksi. Tehtävässä pitää kuitenkin käyttää vektorin p järjestyslukunotaatiota, jotta tehtävä toimisi sellaisenaan yleisellä kolmialkioisen vektorin syötteellä. Kirjainten tilalla käytetään siis termejä $p(1)$, $p(2)$ ja $p(3)$ vastaavasti.

Toiseksi täytyy osata tulkita valmiita funktion rivejä ja erityisesti yksinkertaisia ehtolauseita niin että ymmärtää, mitä funktiossa tapahtuu. Seuraavana on selvitettävä, mihin lukuarvoon diskriminantin laskettua arvoa on verrattava, jotta funktio ilmoittaisi käyttäjän antamasta polynomista reaalisten juurien lukumäärän oikein. Diskriminantin määritelmän mukaanhan oikea vertailtava arvo on 0.

STACK-tehtävän toinen kohta on siinä mielessä puutteellinen, että sen pystyy päättelemään oikeaksi, vaikkei MATLABiin kirjoittaisi riviäkään skriptaa. Toisaalta tehtävän sisältöä on hieman haasteellista laatia kaikille sopivan helpoksi siten että opiskelijan olisi käytännössä pakko kirjoittaa ohjelma MATLABilla. Siinäkään mielessä tehtävä ei täysin parhaalla mahdollisella tavalla tue MATLABin taitojen oppimista, koska annettujen funktion rivien kopioiminen editoriin ja funktion ajaminen sen jälkeen ei aiheuta mitään muuta kuin virheilmoituksen. Opiskelijan täytyy siis itse päätellä alusta alkaen, miten ohjelman saa toimimaan. Funktio voisi toimia jollain tavalla puuttellisesti mutta kuitenkin jatkosta vinkkejä antaen. Toisaalta pelkästään se, että opiskelija tämän tehtävän ratkaistessaan osoittaa ymmärtävänsä, mitä valmiit koodirivit sanovat, on myös hyvä taito jatkoa ajatellen. Kolmannen tehtävän löytää kokonaisuudessaan liitteestä 3.

Tehtävä 4: MATLABin omat funktiot, kuten polyfit, sovitteet ja pienimmän neliösumman menetelmä

Tämä tehtävä sisältää eniten oppiainerajat ylittävää materiaalia, ja moni opiskelija koki nimenomaan tämän tehtävän vaikeimmaksi näistä viidestä tehtävästä. Tehtävän a-kohta ei itse asiassa liity MATLABin käyttöön juurikaan vaan siinä on hyödynnettävä matriisilaskentaa. Tehtävän yhdistäminen matriisilaskentaan on käyttökelpoinen idea, koska tämän kokonaisuuden suorittaminen ajoittuu samalle ajanjaksolle kuin opiskelijoiden ensimmäinen matriisilaskennan kurssi. Syksyllä 2015 tosin MATLABin alkeet tuli olla suoritettuna melko aikaisessa vaiheessa matriisilaskennan kurssia, joten monet opiskelijat eivät ehtineet välttämättä kunnolla sisäistää tehtävään vaadittavaa matriisitulon määritelmää.

Tehtävän a-kohdassa on annettu vektorit x ja y, joista jälkimmäisen arvot ovat satunnaisia kokonaislukuja ja ensimmäisen arvot ovat kiinteitä. Vektoreita vastaavat pisteet on myös piirretty tehtävänannossa näkyvään koordinaatistoon. Tehtävänä on selvittää toisen asteen polynomin $p(x) = ax^2 + bx + c$ sovitteen laskeminen. Toisen asteen polynomin sovitteen voi ratkaista lineaarisella yhtälöryhmällä seuraavasti:

$$A \begin{bmatrix} a \\ b \\ c \end{bmatrix} = y,$$

missä A on 5×3-matriisi, jonka alkiot saadaan annetun vektorin x avulla. Tehtävän vastauksena tarvitsee selvittää matriisin A alkiot ja täyttää ne tyhjiin kohtiin. Jos vastaa oikein, niin tehtävä antaa palautteena valmiit MATLAB-rivit, joilla edelliset muuttujat voi kopioida MATLABiin. Lisäksi tehtävä myös piirtää alussa olevaan JSXGraph-koordinaatistoon sovitteen pisteille, jos a-kohtaan vastaa oikein. Kaikista tehtäväpaketin tehtävistä tässä kohdassa vaaditaan opiskelijalta eniten muuta osaamista kuin MATLAB-osaamista. Tämä tehtävä tukee erityisen paljon opiskelijoiden tutkivaa oppimistyyliä, koska asia voi entuudestaan olla hyvinkin vieras. Tarjolla olevalla ohjemateriaalilla ja varsinkin matematiikan kurssin muulla oppimateriaalilla tehtävästä kuitenkin pystyy selviämään. Lisäksi tehtävä antaa palautteena lisävinkkiä, jos vastaa väärin.

Neljännen tehtävän b-kohta on kaksiosainen. Opiskelijan tulee ratkaista edellä määritellyn sovitepolynomin $p(x)$ kertoimien a, b ja c lukuarvot MATLABilla kahdella eri tavalla. Toinen käytetty tapa on oikealta kertomalla eli komento $A \backslash y$ ja toinen on sisään rakennettu `polyfit`-komento. Ensimmäisessä tapauksessa vastaus on kolmen alkion mittainen pystyvektori ja toisessa tapauksessa vastaus on puolestaan kolmen alkion mittainen vaakavektori. Molemmissa on siitä huolimatta samat alkiot. Tehtävän toteuttaminen on mahdollinen, sillä MATLABin laskentatarkkuudella alkiot ovat samat. Tehtävä osaltaan tukee opiskelijan keksivää oppimista, koska tällainen havainto saattaa antaa opiskelijalle oivalluksen, että nämä kaksi laskentatapaa ovatkin itse asiassa sama asia. Se myös voi joillakin opiskelijoilla toimia niin, että opiskelija osaa itse päätellä vastauksen toiseen kohtaan. Tehtävä neljä löytyy liitteestä 4.

Tehtävä 5: Datan tuominen ulkopuolisesta tiedostosta ja sen käsittely MATLABilla

Tehtäväpaketin viidennessä tehtävässä opiskelijan on määrä ymmärtää, miten MATLABilla voidaan lukea ulkopuolisia tiedostoja ja käsitellä niitä ohjelmistoon sopivina muuttujina. Tehtävän oheismateriaalina on annettu 100×100-kokoinen exceltaulukko, jonka jokaisessa solussa on mielivaltainen desimaaliluku, jonka arvo on reaalilukuvälillä $(1, 100)$. Tiedosto on tarkoituksella isokokoinen ja lukuarvoiltaan satunnainen, jotta opiskelijat eivät yrittäisikään käyttää tehtävän ratkaisemiseen mitään muuta kuin MATLABia. Tällä pyritään siis ikään kuin pakon edessä opettamaan opiskelijoille juurikin MATLABin käyttöä. Tiedosto on xlsx-muotoinen muun muassa siksi että exceltiedostojen käsittely on monella opiskelualalla melko yleistä ja siksi että numeeriset tiedostot kääntyvät MATLABin muuttujiksi melko kivuttomasti. Pyrkimyksenä on saada myös opiskelijat ymmärtämään xlsx-tiedostojen muuntuminen MATLABin matriisimuotoisiksi muuttujiksi, joille pätee muunnoksen jälkeen matriisien laskutoimitukset. Tehtävä on myös selkeästi muita lyhyempi ja vanhoja asioita kertaava. Tällä pyritään auttamaan opiskelijoita läpäisemään pakollinen MATLABin alkeiden suoritettava osuus erityisesti hieman muita haasteellisemman neljännen tehtävän jälkeen. Ei myöskään pidetty välttämättömänä sitä, että tehtävä pyrkisi tarkoituksella olemaan erityisen haastava, koska tehtävän olennaisen aineksen voi oppia helpommalla tehtävänannollakin.

Ensimmäinen puolikas tehtävästä kertaa vanhaa asiaa. Tehtävänä on ensin tallentaa ohjeistuksen avulla exceltiedosto MATLABilla numeeriseksi matriisiksi ja sen jälkeen hakea tästä matriisista järjestyksessä tietty alkio ja ilmoittaa tämä alkio vastauksena. Tämä järjestysalkio on satunnainen kokonaisluku väliltä $(1, 20)$. STACK-tehtävän toiminnallisuuden kannalta riittää muodostaa taulukko, jossa on 20 exceltiedoston ensimmäistä lukuarvoa, ja sen lisäksi muuttuja, joka arpoo luvun yhden ja kahdenkymmenen väliltä. Oikea vastaus on sitten luonnollisesti taulukon järjestyksessä edellä arvottu luku.

Jälkimmäinen puolikas tehtävästä sisältää excel-taulukkoon sisältyvän determinanttilaskun. Lasku on tehtävässä ilmaistu MATLAB-komentona seuraavasti:

```
det(2*A(5:2:15,5:2:(5+10)))
```

Tämäkin tehtävä on tarkoituksella mahdollisimman mutkikkaasti muodostettu lauseke, jotta opiskelijat eivät harkitsisi laskutoimituksen tekoa millään muulla työkalulla kuin MATLABilla. Tehtävän päämääränä on syventää ymmärrystä, että exceltiedosto on muunnettu MATLABin numeeriseksi matriisiksi, jolloin sitä voidaan käsitellä matriisimuotoisena muuttujana ja tehdä matriiseille tuttuja laskutoimituksia. Viides tehtävä on liitteessä 5.

4.4 Tehtävien oheismateriaali

4.4.1 Opetusvideot

Kuten monissa luvussa 3 esitellyissä laskentaohjelmistojen oppimateriaaleissa oli tehty, myös TTY:n uudistetussa MATLABin alkeiden kokonaisuudessa päätettiin kokeilla screencast-tyyppisiä opetusvideoita MATLABin käytöstä. Luvun 2 mukaan tämäntyyppisten videoiden avulla on kätevää opettaa juurikin tietokoneohjelmien käyttöä, koska videokuvan voi keskittää ainoastaan tietokoneruutuun ja esimerkiksi puhetta voi nauhoittaa ulkoisella tai sisäänrakennetulla mikrofonilla. Useimmat muista materiaaleista löydetyt opetusvideot olivat rakenteeltaan sellaisia, että niissä oli nauhoitettu ohjelmiston käyttöä ja puhetta siitä mitä tehtiin samanaikaisesti. MATLABin alkeiden opetusvideot tehtiin samalla idealla, jolloin pystyttiin hyvin havainnollistamaan MATLABin toimintoja käytännössä.

Kuvaruutunauhoitteita voi tehdä monilla erilaisilla sovelluksilla. Nämä videot tehtiin Mac-tietokoneelta valmiiksi löytyvällä QuickTime Player -sovelluksella, ja ääninauhoitetta varten riitti yksinkertaisesti ulkoinen mikrofoni. Näillä yksinkertaisilla menetelmillä sai riittävän hyviä opetusvideoita, joissa kirjoitettiin MATLAB-komentoja samanaikaisesti kuin puhuttiin mitä tehdään. Etuna muihin jo valmiina löytyviin materiaaleihin oli se, että nämä videot olivat suomen kielellä puhuttuja. Useimmat muut videot aiheesta ovat englanniksi. Kaiken kaikkiaan videoissa käsiteltiin enimmäkseen materiaalia, joka liittyi tehtäviin siten että ne pystyi ratkaisemaan, mutta paljon käytiin läpi myös asioita, joita tehtävien ratkaisemiseen ei välttämättä tarvinnut. Opetusvideot tarjoavat havainnollistavan vaihtoehtoisen oppimateriaalin MATLABin alkeille.

4.4.2 Ohjepdf-tiedostot

Jokaisen tehtävän oheen tehtiin myös aiheeseen liittyvä ohjepdf-tiedosto, jossa käsiteltiin tehtävän aihepiiriä sekä myös hieman ylimääräistäkin asiaa. Pääosin tiedostot olivat tekstiä, mutta joitakin kuvia oli myös esimerkiksi MATLABin ikkunanäkymistä, ja kaikki MATLAB-komennot erotettiin selkeästi muusta tekstistä käyttämällä yhtälösisennystä ja ympäröimällä komennot tummankeltaisilla laatikoilla ja kirjoittamalla teksti punaisella. Tiedostoissa esimerkit rakennettiin vaiheittain niin, että opiskelijat pystyivät seuraamaan, mitä mikäkin kirjoitettu komento MATLABissa tekee.

Pdf-tiedostojen lisääminen Moodleen on helppoa eivätkä ne kuormita vielä merkittävästi muun informaation kanssa oppimisympäristöä siten että ne tekisivät siitä sekavan ja informaation osalta ylitsevuotavan. Pdf-ohjemateriaali tukee osaltaan tutkivaa oppimista vaiheittain etenevillä esimerkeillään, koska tällöin opiskelijat pääsevät itse tekemään havaintoja siitä mitä asioita tekemällä mitäkin tapahtuu. Ohjepdf-tiedostojen pituus pidettiin mahdollisimman tiiviinä eli maksimissaan kahdessa tai kolmessa sivussa, jotta liikaa sisäistettävää asiaa ei olisi. Nämä tiedostot ovat kätevä toisenlainen tapa opetella asia. Käytetyt pdf-tiedostot löytyvät liitteistä 6-10.

4.4.3 EXAM-tentti

EXAM on TTY:n sähköinen tenttijärjestelmä. MATLABin alkeita varten luotiin myös EXAM-tentti, jolla ensinnäkin testattiin sähköisen tenttimisen mahdollisuuksia ja lisäksi tentti toimi vaihtoehtoisena suoritustapana niille opiskelijoille, jotka eivät aikamääreeseen mennessä saaneet tehtäviä hyväksytysti suoritettua. Sähköisen tentin kokeilu oli ylipäätään mahdollista siksi että tenttikoneille oli valmiiksi asennettu MATLAB.

Tentti rakennettiin niin, että se testasi kaikkia asioita, mitä Moodlen muullakin materiaalilla testattiin. Tenttijän tulee tentissä ladata pakattu kansio, jossa on kaksi MATLABin p-tiedostoa ja yksi m-tiedosto. M-tiedostoon kirjoitetaan käytännössä tenttivastaus, joka myös palautetaan tentin yhteydessä tarkastettavaksi, ja p-tiedostoista selviää tentin tehtävänanto. P-tiedostoista toinen sisältää tehtävänannon, joka tulostuu MATLABin komentoikkunaan sen jälkeen kun tenttijä on henkilökohtaista opiskelijanumeroaan käyttäen ajanut tiedoston. Tehtävänannon mukaisesti tenttivastaukset kirjoitetaan m-tiedostoon, ja samat vastaukset myös tarkistetaan toisella p-tiedostolla. Tämäkin p-tiedosto ilmoittaa ajon jälkeen komentoikkunaan, onko jossain vastauksessa huomauttamista. Tentissä testataan ulkopuolisen tiedoston lataamista MATLABiin, samaisen tiedoston käsittelyä niin että siitä pitää muodostaa vektori- tai matriisimuotoisia muuttujia, näiden vektorien piirtämistä plot-kuvaajaan datapisteinä sekä sovitteen piirtämistä pisteille.

Sähköistä tenttiä kokeiltiin syksyllä 2015 pelkästään varavaihtoehtona, mutta tätä alustaa pystyy hyödyntämään kyllä enemmänkin. Tällainen sähköinen tentti muun muassa testaa varmemmin MATLABin käsittelytaitoja, koska edellisessä luvussa kuvatut STACK-tehtävät voi periaatteessa pystyä ratkaisemaan ilmankin MATLABia.

4.4.4 Vuorovaikutuksellinen opetus

Luvun 2 mukaisesti verkko-oppimisympäristössä tarjottiin myös vuorovaikutuksellista apua tehtävien ratkaisemiseen. Moodle-sivulla oli käytössä keskustelupalsta, jossa opiskelijat voivat kysyä epäselvistä kohdista. Lisäksi alkeiden vastuuhenkilö tarjosi sähköpostiosoitteensa käyttöön, jotta opiskelijat pystyivät sinne ottamaan yhteyttä tarvittaessa. Nämä ominaisuudet on hyvä lisätä Moodle-toteutuksille, koska hyvin useissa tilanteissa tehtävissä saattaa olla jokin vika, jonka siellä ei kuuluisi olla ja jonka opiskelijat voivat helpommin huomata.

Kokonaan tätä kokonaisuutta ei sähköistetty, vaan tarjottiin myös perinteistä opetusta alkeiden opiskelijoille. Yhtenä viikkona järjestettiin ohjattuja harjoitustilaisuuksia, joihin opiskelijat saivat tulla paikan päälle ratkaisemaan tehtäviä tietokoneilla. Tällainenkin mahdollisuus oli järkevää antaa, koska tällä tuetaan hyvin erilaisia oppimistyylejä.

5. KLUSTERIANALYYSI

5.1 Hierarkkinen klusterointi

Klusterianalyysi on monimuuttujatilastoanalyysimenetelmä, jossa päämääränä on jakaa dataa ryhmiin eli klustereihin samankaltaisuutensa perusteella. Hierarkkinen klusterointi lähtee siitä, että jokainen havainto on aluksi oma klusteri ja samanlaiset havainnot yhdistetään yhdeksi klusteriksi ja ratkaistaan kuinka kaukana uusi klusteri on muista klustereista. Näin jatketaan kunnes kaikki havainnot ovat samassa klusterissa. [31, s. 407-411] Klusterit erotellaan toisistaan yleensä dissimilaarisuuden perusteella eli etäisyysmitalla. Olkoot $\mathbf{x}_i$ ja $\mathbf{x}_j$ datavektoreita. Dissimilaarisuudella d on seuraavat kolme ominaisuutta:

1. $d(\mathbf{x}_i, \mathbf{x}_j) \geq 0$
2. $d(\mathbf{x}_i, \mathbf{x}_i) = 0$
3. $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$

Jos dissimilaarisuus edellisten ehtojen lisäksi toteuttaa kolmioepäyhtälön $d(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_k) + d(\mathbf{x}_k, \mathbf{x}_j)$ ja on nolla täsmälleen silloin kuin $\mathbf{x}_i = \mathbf{x}_j$, niin dissimilaarisuus täyttää metriikan aksioomat [32, s. 1-2]. Yleisesti ottaen dissimilaarisuuden ei tarvitse olla metriikka. Dissimilaarisuuden suuruuden voi määritellä usein eri tavoin mutta tässä työssä määritellään ainoastaan yleisimmin käytetyt euklidinen dissimilaarisuus ja Manhattan cityblock -dissimilaarisuus. Olkoot vektorit $\mathbf{x}_i = (x_{i1}, \ldots, x_{ir})^T$ ja $\mathbf{x}_j = (x_{j1}, \ldots, x_{jr})^T$. Euklidisen dissimilaarisuuden laskukaava on seuraava: [31, s. 412-413]

$$d(\mathbf{x}_i, \mathbf{x}_j) = [(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)]^{1/2}. \tag{5.1}$$

MATLAB käyttää euklidisen dissimilaarisuuden laskemisessa neliöllistä muotoa. Vastaavasti Manhattan cityblock -dissimilaarisuuden laskukaava menee samoilla vektoreilla $\mathbf{x}_i$ ja $\mathbf{x}_j$ seuraavasti: [31, s. 413]

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{r} |x_{ik} - x_{jk}|.$$

Käsitellään seuraavaksi agglomeratiivisen hierarkkisen klusteroinnin algoritmi. Tämä käsitellään nimenomaan siksi että MATLABin sisäänrakennettu klusterointilaskentakomento käyttää tätä algoritmia ja lisäksi myöhemmin tässä työssä Moodlen lokitietodataa on nimenomaan klusteroitu tätä laskentamenetelmää hyväksi käyttäen. Algoritmi on seuraava: [31, s. 415]

1. Olkoon syötteenä joukko $L = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, missä n on alkuperäinen määrä klustereita tai havaintoja.
2. Laske $D = (d_{ij})$, joka on $n \times n$-kokoinen dissimilaarisuusmatriisi n klusterin välillä. $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, 2, \ldots, n.$
3. Etsi matriisista D pienin dissimilaarisuus d_{ij}. Yhdistä klusterit i ja j uudeksi klusteriksi ij.
4. Laske dissimilaarisuudet $d_{ij,k}$ uuden klusterin ij ja kaikkien muiden klusterien $k \neq ij$ välillä. Nämä dissimilaarisuudet riippuvat siitä mitä linkitysmenetelmää käyttää. Niitä on normaalisti enemmänkin vaihtoehtoina mutta käydään tässä tapauksessa läpi ainoastaan yksittäislinkitys (single linkage): $d_{ij,k} = \min(d_{i,k}, d_{j,k})$.
5. Muodosta uusi $(n-1) \times (n-1)$- kokoinen dissimilaarisuusmatriisi D poistamalla vaaka- ja pystyrivit i ja j ja lisäämällä uuden vaaka- ja pystyrivin ij, jonka dissimilaarisuudet laskettiin edellisessä askeleessa.
6. Toista askeleet 3, 4 ja 5 kaiken kaikkiaan $n-1$ kertaa. Loppujen lopuksi kaikki lähtöarvot ovat yhdistyneet yhdeksi klusteriksi.
7. Ulostulona on lista siitä, mitkä klusterit ovat yhdistyneet milläkin askeleella ja jokaisen yhdistymisen dissimilaarisuusarvo.

Näytetään yksi yksinkertainen esimerkki hierarkkisen klusteroinnin algoritmin vaiheista. Olkoot vektorit $\mathbf{x}_1 = (0{,}0)^T$, $\mathbf{x}_2 = (1{,}0)^T$, $\mathbf{x}_3 = (3{,}0)^T$ ja $\mathbf{x}_4 = (3{,}3)^T$. Nämä vektorit ovat nyt joukon L alkiot. Seuraavaksi lasketaan algoritmin toisen askeleen mukaisesti dissimilaarisuusmatriisi D käyttäen euklidista dissimilaarisuusmittaa yhtälön (5.1) mukaisesti:

$$D_{(1)} = \begin{array}{c} d_{ij} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 3 & \sqrt{18} \\ & 0 & 2 & \sqrt{13} \\ & & 0 & 3 \\ & & & 0 \end{pmatrix} \end{array}.$$

Seuraavaksi siirrytään algoritmin kolmanteen vaiheeseen ja etsitään matriisista D pienin dissimilaarisuus d_{ij}. Tässä tapauksessa se on $d_{12} = d_{21} = 1$. Yhdistetään vektorit $\mathbf{x}_1$ ja $\mathbf{x}_2$ uudeksi klusteriksi numero 12. Lasketaan uuden klusterin ja kaikkien muiden klusterien väliset uudet dissimilaarisuudet käyttäen yksittäislinkitystä. Tämä on algoritmin neljäs vaihe. Tällöin uudeksi dissimilaarisuusmatriisiksi D saadaan algoritmin viidennen vaiheen mukaisesti:

$$D_{(2)} = \begin{array}{c} d_{ij} \\ 12 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 12 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 2 & \sqrt{13} \\ & 0 & 3 \\ & & 0 \end{pmatrix} \end{array}.$$

Näin algoritmiä toistetaan uudelleen kolmannesta vaiheesta alkaen, kunnes kaikki klusterit ovat yhdistyneet yhdeksi klusteriksi:

$$D_{(3)} = \begin{array}{c} d_{ij} \\ 123 \\ 4 \end{array} \begin{array}{c} \begin{array}{cc} 123 & 4 \end{array} \\ \begin{pmatrix} 0 & 3 \\ & 0 \end{pmatrix} \end{array}.$$

Viimeinen klusterien yhdistyminen esimerkissä on klusterien 123 ja 4 yhdistyminen, jonka dissimilaarisuus on $d_{123,4} = 3$.

5.2 K-means -klusterointi

Epähierarkkisessa klusteroinnissa data jaetaan ennalta määrättyyn määrään K eri klustereita eikä näillä klustereilla ole hierarkkista riippuvuutta toisistaan. Yksi usein käytetty epähierarkkisen klusteroinnin menetelmä on k-means -klusterointi. Menetelmän algoritmi perustuu ennalta valittuihin klustereihin ja niiden painopisteisiin (centroid) sekä klustereiden väliseen mielivaltaiseen dissimilaarisuusmittaan d ja virheiden neliösummaan (ESS, error sum of squares). K-means -klusteroinnin algoritmi menee kokonaisuudessaan seuraavalla tavalla: [31, s. 422-424]

1. Olkoon syötteenä joukko $L = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$ ja olkoon K klusterien määrä.

2. Tee toinen seuraavista:

   - Muodosta lähtökohtaisesti satunnaisesti joukon L alkiot klustereihin K ja laske jokaiselle klusterille k sen painopiste $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
   - Määritä ennalta K kappaletta klusteripainopisteitä $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.

3. Laske jokaiselle joukon L alkiolle neliöity euklidinen etäisyys alkion sen hetkisen klusteripainopisteen kanssa: $ESS = \sum_{k=1}^{K} \sum_{c(i)=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^T (\mathbf{x}_i - \bar{\mathbf{x}}_k)$, missä $\bar{\mathbf{x}}_k$ on k:s klusteripainopiste ja $c(i)$ on klusteri, joka sisältää alkion $\mathbf{x}_i$.

4. Järjestä jokainen joukon L alkio uudelleen sen lähimmälle klusteripainopisteelle, jotta luvun ESS suuruus laskee. Päivitä klusteripainopisteet uudelleen uudelleenjärjestelyn jälkeen.

5. Toista askeleet 3 ja 4 niin pitkään kunnes uudelleenjärjestelyä ei enää tarvita.

Sekä hierarkkista klusterointia että k-means -klusterointia hyödynnetään seuraavassa luvussa opiskelijoiden vastauskäyttäytymiseen, minkä pohjalla on tässä luvussa esitetty matemaattinen tausta. Laskennat on toteutettu MATLABilla, jolla laskenta-algoritmit ovat samoja kuin yllä. Erityisesti hierarkkisessa klusteroinnissa on otettu käyttöön ainoastaan ne vaihtoehdot, joita MATLABilla analysoitaessakin käytetään.

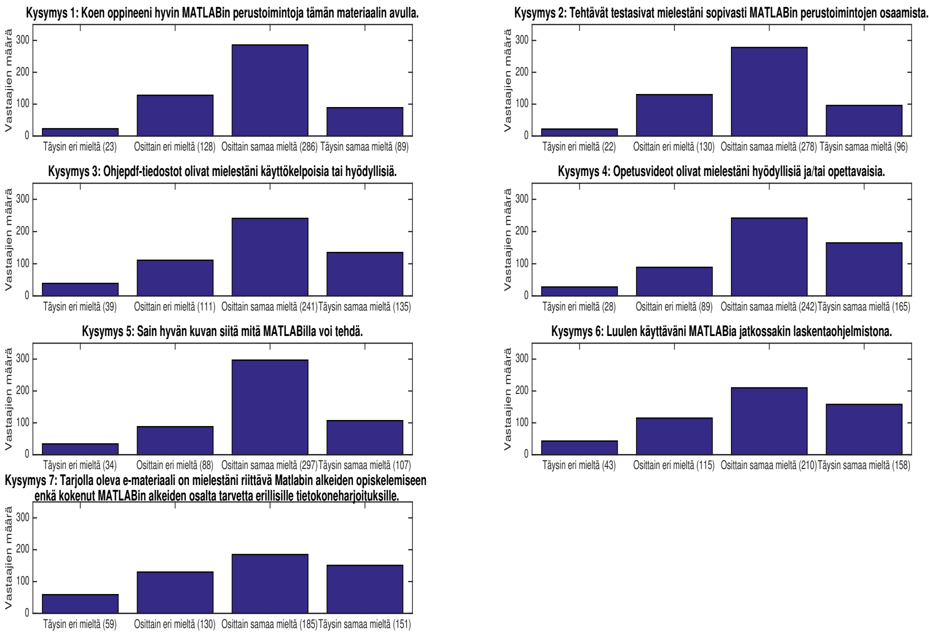
6. MATLABIN ALKEIDEN TOTEUTUS TTY:LLÄ

6.1 Opiskelijoilta kerätty palaute materiaalista

Kaikkia alkeet suorittaneita opiskelijoita kannustettiin vastaamaan samalta Moodle-sivulta löytyneeseen erilliseen kyselyyn. Kyselyllä kerättiin opiskelijoilta sekä kvantitatiivista että kvalitatiivista palautetta, jotta MATLABin alkeita voitaisiin kehittää jatkoa ajatellen. Bonuksena kaikille vastanneille annettiin kyselyyn vastaamisesta yksi ylimääräinen piste, joka laskettiin läpipääsypistemäärään mukaan. Kyselyssä oli seitsemän monivalintakysymystä, joissa pyydetiin arvioimaan väittämiä vaihtoehdoilla täysin eri mieltä, osittain eri mieltä, osittain samaa mieltä sekä täysin samaa mieltä. Näiden seitsemän kysymyksen lisäksi oli kahdeksas avoin kysymys, johon sai kirjoittaa vapaamuotoista palautetta. Kyselyssä esitetyt kysymykset olivat seuraavat:

1. Koen oppineeni hyvin MATLABin perustoimintoja tämän materiaalin avulla.
2. Tehtävät testasivat mielestäni sopivasti MATLABin perustoimintojen osaamista.
3. Ohjepdf-tiedostot olivat mielestäni käyttökelpoisia tai hyödyllisiä.
4. Opetusvideot olivat mielestäni hyödyllisiä ja/tai opettavaisia.
5. Sain hyvän kuvan siitä mitä MATLABilla voi tehdä.
6. Luulen käyttäväni MATLABia jatkossakin laskentaohjelmistona.
7. Tarjolla oleva e-materiaali on mielestäni riittävä MATLABin alkeiden opiskelemiseen enkä kokenut MATLABin alkeiden osalta tarvetta erillisille tietokoneharjoituksille.
8. Vapaa palaute, parannusehdotukset: (avoin kysymys)

Kysymyksiä ei tarkoituksella ollut montaa, koska tuon määrän koettiin antavan riittävästi tietoa alkeiden sisällöstä. Kysymyksillä pyrittiin kartoittamaan paitsi mielipiteitä opetusmateriaalista myös mielipiteitä MATLABin käytöstä ylipäätään. Yksi toteutuksen tavoitteista oli kuitenkin edistää MATLABin käytön yleistymistä opiskelijoilla. Kyselyyn tuli vastauksia kaiken kaikkiaan 677 kappaletta, mikä on hyvä lukema. Lukema on suunnilleen sama kuin alkeet suorittaneiden opiskelijoiden kokonaislukumäärä. vastauksista koottiin kuvan 6.1 mukaiset pylväsdiagrammit.



Kuva 6.1 *Jokaista palautekysymystä vastaava pylväsdiagrammi opiskelijoiden kvantitatiivisista vastauksista.*

Kyselyn kvantitatiivisessa analyysissä ei erityisen isoja ihmeitä näkynyt. Suurin osa kyselyyn vastanneista oli samaa mieltä väittämien kanssa eli kokivat uudistetun e-oppimateriaalin hyödylliseksi MATLAB-taitojen oppimiseen. Kysymyksessä 7 oli suhteessa useampi vastaaja eri mieltä väittämän kanssa, eli monet kokivat e-oppimateriaalin kuitenkin hieman riittämättömäksi MATLAB-taitojen kunnolliseen oppimiseen. Esimerkiksi kontaktiopetuksen lisäämistä voisi harkita.

Kyselyvastausten kvalitatiivisessa analyysissä eli avoimen kahdeksannen kysymyksen vastauksista nousi esille muutamia selkeitä seikkoja useassa vastauksessa. Kaikki kyselyyn vastanneet eivät avointa palautetta kirjoittaneet, ja merkittävä osa kirjoitti olevansa tyytyväinen materiaaliin, mutta hyvin monet opiskelijat antoivat kokonaisuudelle parannusehdotuksia. Erityisen monessa vastauksessa todettiin, että tehtävät ovat liian paljon pelkkää MATLAB-komentojen kopioimista ja liittämistä ja että ohjelman käyttöä oppisi paremmin, jos tehtävissä joutuisi miettimään enemmän. Sitä vastoin vähän pienempi osa opiskelijoista piti tehtäviä joko sopivan haastavina tai liian haasteellisina. Neljännen tehtävän a)-kohtaa moni piti liian vaikeana erityisesti siksi että se ei testannut MATLAB-osaamista juurikaan ja siksi että alkeiden suorittamisajankohta ajoittui hieman epäedullisesti toisen matematiikan peruskurssin kanssa, ja matriisit eivät olleet intuitiivisella tasolla kovin tuttuja monelle. Lähes kaikki opetusvideoita kommentoineet pitivät niitä hyvänä oheismateriaalina. Ohjepdf-tiedostot keräsivät hieman ristiriitaista palautetta. Osa piti niitäkin hyödyllisinä ja selkeinä, osa taas suppeina ja hyödyttöminä. Monessa vastauksessa toistui, että MATLABin syvällinen oppiminen jää tämän materiaalin pohjalta hieman vajaaksi. Suuri osa olisi myös kaivannut enemmän ohjattuja harjoituksia ja kontaktiopetusta e-oppimateriaalin tueksi.

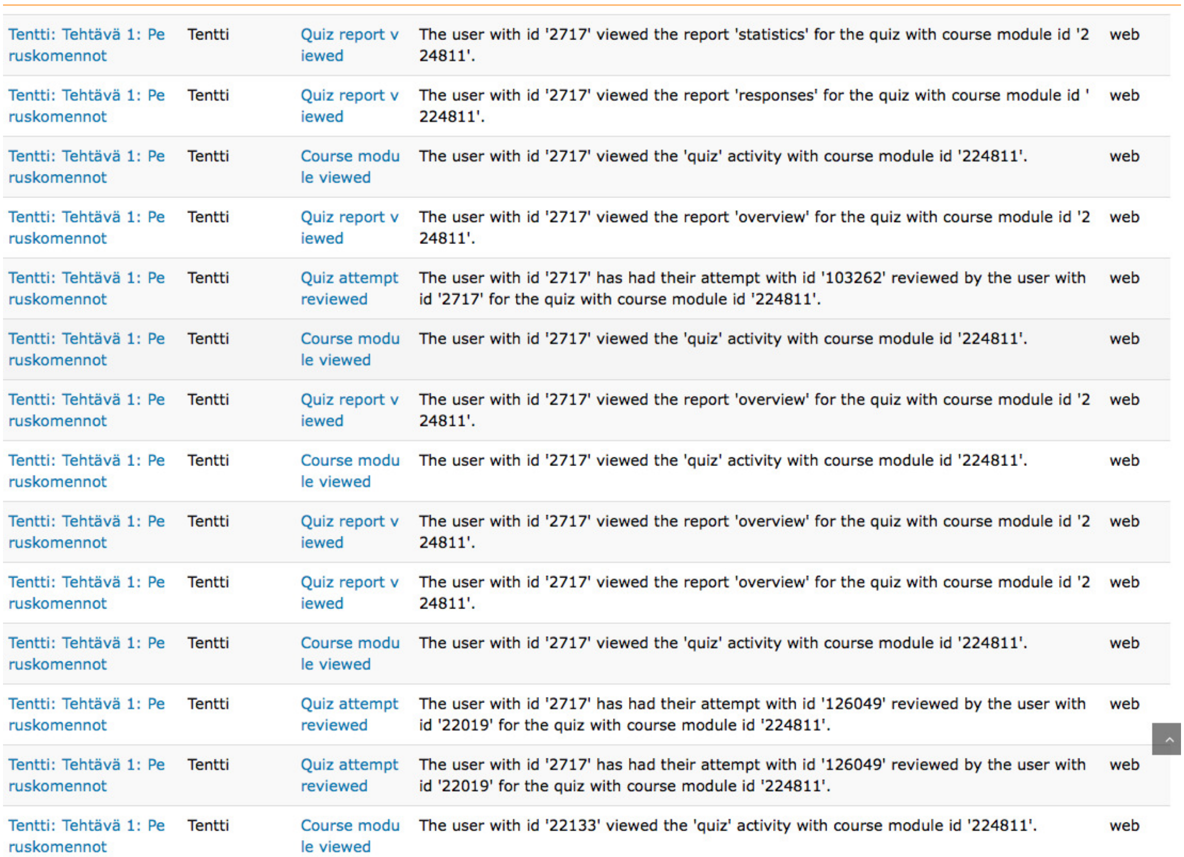
6.2 Moodlen lokidatan klusterianalyysi

6.2.1 Moodlen lokitiedot

Moodle kerää lokitietoja jokaisen opiskelijan suorituksesta. Tämän työn kannalta olennaiset tiedot ovat kertyneet pistemäärät, MATLAB-osion aloitus- ja lopetusajankohdat, suoritukseen kulunut aika sekä tämän datan korrelaatio saman vuoden perustaitojen testien kokonaispistemäärään. Koska Moodle kerää äsken luetelluista asioista lokitietoja, niitä on kätevä analysoida valmiina. Tietojen pohjalta tutkitaan, millaisiin klustereihin syksyn 2015 MATLAB-osion suorittaneet opiskelijat voidaan jakaa luvussa 5 esitellyn klusterianalyysin teorian avulla.

Moodlesta kerätyt lokitiedot sisältävät taulukkomuotoisena tuhansia rivejä dataa. Tämä lokidata sisältää opiskelijoiden tehtävissä tekemiä klikkauksia. Jokaiselle opiskelijalle ja tehtävälle on oma tunnistenumeronsa ja jokaiselle erityyppiselle toiminnalle Moodlessa on myös oma tunnistenumeronsa. Tehtävittäin data jakautui lokirivimäärän perusteella tässä tapauksessa seuraavasti: ensimmäisessä tehtävässä 12966 riviä, toisessa tehtävässä 7963, kolmannessa tehtävässä 11229, neljännessä tehtävässä 9930 ja viidennessä tehtävässä 6843 riviä. Datan voi ladata esimerkiksi xlsx- tai csv-muodossa koneelle. Niitä voi käsitellä mieleisikseen taulukonkäsittelytyökaluilla, jotta analysointi esimerkiksi MATLABilla olisi helpompaa. Kuvassa 6.2 on esimerkkikuva siitä, miltä Moodlen lokitiedot näyttävät Moodlessa katsottuna.

Moodlesta on mahdollista tuoda muutakin tehtävään liittyvää opiskelijakohtaista dataa, kuten arvosanojen keskiarvoja. Tällä kertaa kuitenkin ylhäällä luetellut tiedot riittävät seuraavan kappaleen analyysin kannalta.



Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'statistics' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'responses' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Course module viewed	The user with id '2717' viewed the 'quiz' activity with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'overview' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz attempt reviewed	The user with id '2717' has had their attempt with id '103262' reviewed by the user with id '2717' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Course module viewed	The user with id '2717' viewed the 'quiz' activity with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'overview' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Course module viewed	The user with id '2717' viewed the 'quiz' activity with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'overview' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz report viewed	The user with id '2717' viewed the report 'overview' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Course module viewed	The user with id '2717' viewed the 'quiz' activity with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz attempt reviewed	The user with id '2717' has had their attempt with id '126049' reviewed by the user with id '22019' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Quiz attempt reviewed	The user with id '2717' has had their attempt with id '126049' reviewed by the user with id '22019' for the quiz with course module id '224811'.	web
Tentti: Tehtävä 1: Peruskomennot	Tentti	Course module viewed	The user with id '22133' viewed the 'quiz' activity with course module id '224811'.	web

Kuva 6.2 *Osa Moodlessa näkyvää lokia tehtävässä tehdyistä toiminteista. Jokaiselle opiskelijalle, toiminnolle ja tehtävälle on oma numerotunnisteensa.*

6.2.2 Lokitietojen käsittely ja MATLABilla toteutettu klusterianalyysi

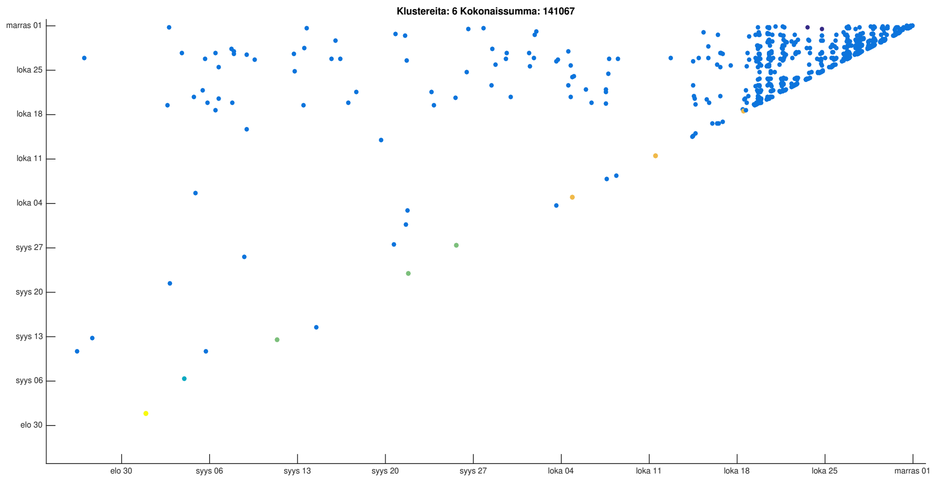
Moodlesta saadut lokitiedot analysoitiin MATLABin klusterointiominaisuuksilla. Analysointia varten jokaisen opiskelijan kohdalta kerättiin ensimmäinen päivämäärä, jolloin kukin opiskelija aloitti tehtäviin vastaamisen sekä lopetuspäivämäärä, jolloin opiskelija sai hyväksytysti suoritettua alkeiden kokonaisuuden. Analyysissä hyödynnettiin Moodlen lokidatan lisäksi opiskelijoille kertyneitä kokonaispistemääriä. Nämä molemmat tiedot yhdistämällä saatiin opiskelijoista jakauma sen mukaan, kuinka kauan heillä kesti läpäistä alkeet hyväksytysti. Klusterianalyysillä haluttiin tutkia opiskelijoiden suorituskäyttäytymistä ja jakaa heidät klustereihin sen perusteella, missä ajassa he tehtävät suorittivat. Vertailun vuoksi erillisistä tiedostoista haettiin myös samojen opiskelijoiden alkusyksystä tekemän matematiikan perustaitojen testin pistemäärät sekä syksyllä pidettyjen kahden ensimmäisen matematiikan peruskurssin arvosanat. Näitä verrattiin klusterianalyysin lopputulokseen ja tutkittiin, millaisia yhtäläisyyksiä MATLABin alkeiden suorituskäyttäytymisessä oli muihin matematiikan opintoihin.

Ennen itse klusterointia lokidata täytyi käsitellä järkevämpään muotoon. Kaikki taulukkomuotoisen datan käsittely ja klusterianalyysi tehtiin MATLABilla. Käytettyä MATLAB- koodia voi kokonaisena tarkastella liitteestä 11. Tämä ei ole ainut mahdollinen keino toteuttaa analyysiä mutta tässä työssä se tehtiin seuraavalla tavalla.

Moodlesta ladattu lokidata on sekaisin numeerista ja merkkijonomuotoista dataa. Lokidata on jaettu yhteen exceltiedostoon, jossa on viidellä eri välilehdellä viiden eri tehtävän tiedot, kuudennella välilehdellä on tehtävistä kertyneet kokonaispisteet opiskelijoittain ja seitsemännellä välilehdellä on vertailun vuoksi opiskelijoiden samansyksyiset matematiikan perustaitojen testin tulokset. Lisäksi erillisessä csv-tiedostossa on myös samojen opiskelijoiden syksyn kahden ensimmäisen matematiikan peruskurssin arvosanat, joita vertaillaan muun lokidatan ohella. Kuuden ensimmäisen välilehden lokidata tallennetaan erikseen `xlsread`-komennolla raakadatasoluksi, jotta se olisi yhtenäistä tietotyypiltään. Tämän jälkeen suodatetaan muodostuneista soluista ei-numeeriset solut pois. Tämä muunnos tarvitaan, koska MATLAB ei osaa käsitellä merkkijonomuodossa olevia numeraaleja numeroina.

Seuraavaksi aletaan muodostaa klusterianalyysiin tarvittavia vektoreja valmiina olevasta käsitellystä datasta. Tässä kohtaa tarpeelliset tiedot, eli opiskelijanumerot ja tapahtumanimet, ryhmitellään `grp2idx`-komennoin. Tämän jälkeen havaitaan, että äsken ryhmitellyissä vektoreissa järjestyksessä kahdeksas alkio sisältää tapahtumanimestä tiedon `Tenttivastaus palautettu`. Haetaan sen jälkeen datasta vain ne rivit, joissa tenttivastaus on palautettu ja lasketaan yhteenvetona tilastolliset parametrit komennolla grpstats. Lopuksi tehdään sekä k-means -klusteroinnilla että hierarkkisella klusteroinnilla kuusi eri testitapausta yhdestä kuuteen klusterilla. Hierarkkiseen klusterointiin käytetään clusterdata-komentoa ja ja molemmissa klusteroinneissa käytetään cityblock-metriikkaa. Nämä algoritmit toimivat vastaavilla tavoilla kuin luvussa 5 esitellyt klusterianalyysin algoritmit. Tämän jälkeen huomataan, että hierarkkisella klusteroinnilla ei oikein saada mielekästä klusterointia, koska se jakaa opiskelijoiden suoritukset liian painottuneesti vain yhteen klusteriin, kuten esimerkiksi kuvassa 6.3 näkyy. Muilla klusterilukumäärillä tilanne on hyvin samankaltainen, minkä voi nähdä liitteestä 11. Hierarkkinen klusterointi onnistui jakamaan näytteet järkevämmin klustereihin sitten kun käytti toisen tyyppistä linkitystä. Täydellisellä linkityksellä (complete linkage) ja keskimääräisellä linkityksellä (average linkage) näytteet jakaantuivat selkeämmin ja tasaisemmin eri klustereihin, mutta opiskelijoiden käyttäytymismallien analysoinnin kannalta k-means -klusteroinnin lopputulos oli kuitenkin mielekkäämpi. Liitteestä 11 löytyy vain yksittäislinkityksen hierarkkinen klusterointi.

K-means -klusteroinnilla saavutetaan järkevämpiä lopputuloksia, koska algoritmi

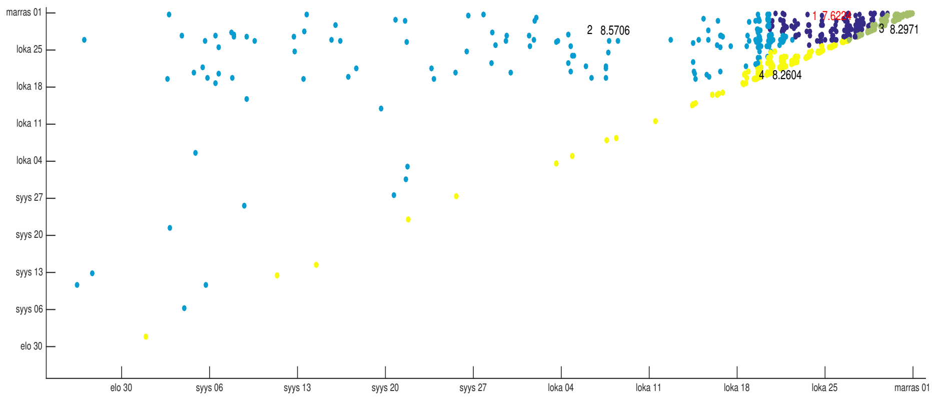


Kuva 6.3 *Hierarkkinen klusterointi datalle kuudella klusterilla. Kuvaaja näyttää vaaka-akselilla kunkin opiskelijan ensimmäisen palautusajankohdan ja pystyakselilla viimeisen palautusajankohdan.*

osaa jakaa opiskelijat tasaisemmin ja järkevämmin eri ryhmiin. Erityisesti havaitaan, että neljällä klusterilla opiskelijat jakautuvat neljään melko selkeään ja loogiseen klusteriin, kuten kuvassa 6.4 näkyy. Tässä klusteroinnissa käytettiin alkutilana MATLABin `kmeans`-komennon tilaa nimeltä `uniform`, joka tuotti useilla eri iteraatioilla yhdenmukaisimmat lopputulokset. Kyseinen alkutila valitsee klusterien määrän mukaisesti pisteet datasta `kmeans`-komennossa määritetyllä etäisyydellä. Myös muita MATLABin `kmeans`-komennon alkutiloja kokeiltiin ja ne tarjosivat suunnilleen vastaavia lopputuloksia. Niissä kuitenkin ilmeni harvemmin samanlaisia klustereita, ja lopputuloksissa oli useammin heittoa. Lopullisena dissimilaarisuusmittana käytettiin `cityblock`-mittaa, joka on etäisyyksien erotusten itseisarvon summa. Myös muita dissimilaarisuusvaihtoehtoja kokeiltiin, mutta ne eivät jakaneet klustereita kovin yhdenmukaisesti, vaan joka kerralla niissä oli selkeitä eroavaisuuksia. Lisäksi muun muassa kosinimitta saattoi jakaa jotkut opiskelijat samaan klusteriin, vaikka kuvaajalla ne olivat toisistaan hyvin erillään.

Neljän klusterin k-means -klusteroinnin neljästä klusterista on lisäksi laskettu keskiarvot ja keskihajonnat kunkin klusterin opiskelijoiden matematiikan kahdesta ensimmäisestä peruskurssista syksyn aikana sekä analysoitu klusterien yhteydessä olleiden perustaitojen testien pistemäärien keskiarvojen tilastollista merkittävyyttä kahden näytteen t-testillä. Klusterianalyysin tulokset nähdään tiivistetysti taulukossa 6.1.

Taulukon 6.1 tuloksista ja kuvasta 6.4 havaitaan, että klusterissa 1 on ne opiskelijat,

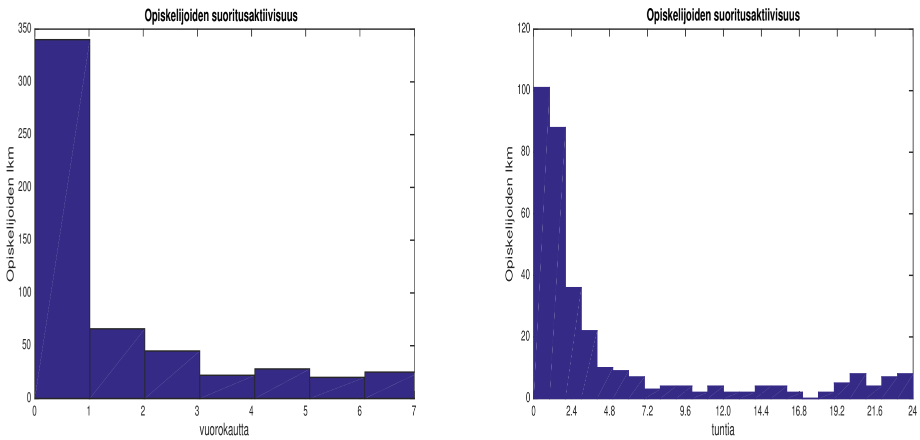


Kuva 6.4 *K-means -klusterointi datalle neljällä klusterilla. Kuvaaja näyttää vaaka-akselilla kunkin opiskelijan ensimmäisen palautusajankohdan ja pystyakselilla viimeisen palautusajankohdan. Klusterien yhteyteen on merkitty klusterin järjestysnumero 1, 2, 3 tai 4 sekä syksyn matematiikan perustaitojen testin keskiarvotulos kyseisen klusterin opiskelijoille.*

Taulukko 6.1 *Moodlen lokidatasta tehdyn klusterianalyysin tuloksien vertailu opiskelijaklustereittain.*

Klusteri (nro)	Opiskelijat (lkm)	Matematiikan perustaitojen testin arvosanakeskiarvo ± keskihajonta	Ensimmäisen matematiikan kurssin arvosanakeskiarvo ± keskihajonta	Toisen matematiikan kurssin arvosanakeskiarvo ± keskihajonta
1	155	$7{,}62 \pm 3{,}21$	$2{,}18 \pm 1{,}60$	$2{,}60 \pm 1{,}59$
2	183	$8{,}57 \pm 3{,}24$	$2{,}70 \pm 1{,}60$	$3{,}03 \pm 1{,}58$
3	177	$8{,}30 \pm 3{,}18$	$2{,}02 \pm 1{,}71$	$2{,}03 \pm 1{,}73$
4	181	$8{,}26 \pm 3{,}28$	$2{,}82 \pm 1{,}59$	$2{,}99 \pm 1{,}62$

jotka aloittavat lähellä harjoitusten loppua mutta kuitenkin melko hyvissä ajoin ja joilla kestää suoritusten loppuun saaminen suunnilleen vuorokausien verran. Klusterissa 2 on opiskelijat, jotka suorittavat alkeet hyvissä ajoin ja kiirettä pitämättä. Klusterissa 3 on opiskelijat, jotka lykkäävät koko alkeiden suorituksen viimeisille mahdollisille päiville. Klusterissa 4 on opiskelijat, jotka suorittavat kokonaisuuden nopeasti heti sen jälkeen kun ovat aloittaneetkin. Perustaitojen testin tuloksen yhteneväisyyttä tarkastellessa huomataan, että klusterin 2 opiskelijat ovat suoriutuneet testistä keskimäärin parhaiten, kun taas huonoiten suoriutuvat klusterin 1 opiskelijat. Klusterien 3 ja 4 opiskelijat ovat suoriutuneet testistä suunnilleen yhtä hyvin. Nämä tulokset eivät varsinaisesti yllättäviä kuitenkaan ole, vaan on odotettavissakin, että rauhassa ja ajoissa tehtävissä suoriutuneet saavat parempia arvosanoja kuin ne jotka jättävät suorittamisen viime tippaan. Testin keskihajonnat eivät myöskään merkittävästi poikkea toisistaan. Matematiikan peruskurssien arvosanoissa klusterin 1 opiskelijat eivät kuitenkaan ole heikoimmin menestyneet, vaan klusterin 3 opiskelijat ovat selkeästi heikoimpia molemmissa kursseissa. Mielenkiintoinen havainto on myös, että parhaiten perustaitojen testissä menestynyt klusteri 2 on toisessa matematiikan peruskurssissa parhaiten menestynyt, kun taas ensimmäisessä kurssissa klusteri jää hieman neljännen klusterin taakse. Klusterien 2 ja 4 opiskelijat ovat kuitenkin matematiikan kursseissa melko tasavahvoja. Klusterianalyysin perusteella voidaan jonkin verran vetää yhtäläisyyksiä MATLABin alkeiden suorituskäyttäytymisen ja muun opintomenestyksen välille. Selkeästi huomataan, että ne opiskelijat, jotka suoriutuvat alkeista joko nopeasti heti aloitettuaan tai hitaasti ja rauhallisesti menestyvät myös muissa opinnoissaan, kun taas ne opiskelijat, jotka venyttävät alkeiden suoritustaan viimeisille päiville, menestyvät muissakin matematiikan opinnoissaan hieman heikommin kuin muut. Liian lopullisia johtopäätöksiä näistä tuloksista ei kuitenkaan kannata vielä vetää, koska tämän työn kirjoittamisen aikaan kaikki syksyn 2015 opiskelijat eivät olleet vielä saaneet arvosanaa matematiikan kursseista. Ensimmäisessä matematiikan kurssissa arvosana puuttui 69 opiskelijalta ja toisessa matematiikan kurssissa 168 opiskelijalta. Nämä tulokset ja johtopäätökset ovat siis enimmäkseen suuntaa antavia, koska erityisesti toisen matematiikan kurssin kohdalla arvosanajakauma voi muuttua merkittävästikin sitten kun loput opiskelijat saavat arvosanan kurssista. Tulokset ja myös muut klusterikuvaajat on nähtävissä liitteessä 11.



Kuva 6.5 *Histogrammi opiskelijoiden alkeiden suoritukseen kuluvasta ajasta vuorokausina ja tunteina.*

Opiskelijoiden suoritusajoista piirrettiin myös MATLABilla histogrammit, jotka osoittavat, että ajallisesti useimmilla opiskelijoilla ei mene yksittäisiä vuorokausia kauempaa kokonaisuuden suorittamiseen. Lisäksi histogrammeista huomaa, että noin kolmasosa kaikista opiskelijoista suorittaa alkeet hyväksytysti alle kolmessa tunnissa. Tehtäviä tätä aikamäärettä kauemmin tekevät ovat selkeässä vähemmistössä. Alkeiden kokonaisuus on siis siinäkin mielessä onnistunut, ettei sen suorittamiseen kulunut valtaosalla liikaa aikaa. Histogrammit löytyvät kuvasta 6.5 ja niiden muodostamisen näkee myös liitteessä 11.

6.3 Tulevaisuuden näkymiä

Syksyllä 2015 saatujen kokemusten perusteella voidaan todeta, että TTY:n peruskurssien yhteydessä opetettavia pakollisia MATLABin alkeita lähdetään kehittämään tästä muodosta, mihin tämän työn kirjoittamisaikana mentiin eikä vanhaan e-oppimateriaalia hyödyntämättömään malliin aiota enää palata takaisin. Automaattisesti itsensä tarkastavat ja itsenäisen oppimisen mahdollistavat STACK-tehtävät ovat merkittävän suuri etu tietokoneluokissa tehtäviin MATLAB-harjoituksiin verrattuna. Tämänhetkinen kokonaisuus tarjoaa hyvän pohjan alkeiden opetuksen kehittämiselle.

Yksi mahdollisuus olisi kehittää tehtävänantoja enemmän oppimisprosessia tukevaan suuntaan. Tällä hetkellä iso osa tehtävänannoista on vielä suurelta osin kopioiliitä -MATLAB-komentoja, joissa MATLABin oppimistaidot eivät välttämättä kehity kovinkaan paljoa eikä ymmärrys tapahtuvasta asiasta merkittävästi kasva. Tehtävänannot voisi siis vastaisuudessa naamioida niin, että niissä lukee sanallisesti se mitä kuuluu tehdä ja opiskelijan tehtäväksi jää selvittää oheismateriaaleista, miten tehtävä toteutetaan MATLAB-komennoin. Esimerkki tästä voisi olla tehtävänantona *Hae MATLABilla matriisin A järjestyksessä yhdeksäs alkio* sen sijaan että kirjoittaisi suoraan *Kopioi MATLABiin A*(9). Syksyllä 2015 opiskelijoilla ei myöskään keskimäärin kulunut kauaa aikaa tehtävien suorittamiseen, joten niiden sisältöjäkin voisi olla mahdollista laajentaa. Tehtävien kehittämisessä tulee kuitenkin noudattaa tiettyä varovaisuutta, ettei sorru tekemään yleisen mielipiteen mukaan liian haasteellisia tehtäviä. Selkeä kehittämiskohde olisi myös alkeiden kokonaisuuden ajoitus, jotta opiskelijoilla olisi mahdollisesti jokin käsitys matriisilaskennasta.

Toinen kehittämisen kohde voisi olla luvussa 4.4.3 mainittu sähköinen tentti. Sen kaikkia hyötyjä ei vielä ehditty valjastaa kunnolla, joten syksyllä 2015 sen toteutus jäi lähinnä varavaihtoehdoksi mattimyöhäisille ja heikosti alkeissa menestyneille. Tentti kuitenkin tenttitehtävän luonteesta riippuen mahdollistaisi oikeasti itsenäisten MATLAB-taitojen oppimisen, sillä tentissä ei ole muita apuvälineitä. P-tiedostoilla pystyy kuitenkin luomaan melko helposti MATLABin alkeiden oppimäärän mukaisen tentin, joka testaa taitoja riittävän kattavasti.

7. YHTEENVETO

Tämä työ käsitteli TTY:llä syksyllä 2015 pilotoitua MATLABin alkeiden kokonaisuutta, joka oli pakollinen suoritettava osa ensimmäisen vuoden opiskelijoiden toista perusmatematiikan kurssia. Työ eteni yleisemmästä taustatarkastelusta tarkempaan taustoitukseen. Ensin käsiteltiin e-oppimisen ja e-oppimateriaalien pedagogisia taustoja, minkä jälkeen käytiin läpi meillä ja muualla maailmalla olevien laskentaohjelmistojen sisältöjä. Niistä haettiin ideoita ja yhteneväisyyksiä oman toteutuksen välille, jotta voitiin oman alkeiden kokonaisuuden oppimateriaalisisältö perustella. MATLABin alkeiden sisällön toteutusmenetelmät ja sisällön kytkeytyminen niin pedagogiseen taustateoriaan kuin muihin laskentaohjelmisto-oppimateriaaleihin käsiteltiin yksityiskohtaisesti. Loppujen lopuksi käsiteltiin itse alkeiden syksyn 2015 toteutusta, sen onnistumista ja havaintoja.

Opiskelijoiden harjoitusten suorituksista tehdystä klusterianalyysistä voidaan tehdä jatkokehittelyä varten päätelmiä opiskelijoiden harjoituskäyttäytymisestä. Suurin osa suorituksista painottui harjoitusjakson loppupäähän, mikä tietenkin on normaalia lähes kaikissa opintoihin liittyvissä projekteissa, mutta osaltaan ilmiö voi tässä tapauksessa johtua myös alkeiden ja matematiikan peruskurssien sisältöjen osittaisella kohtaamattomuudella. Monet MATLABin alkeiden tehtävistä käsittelivät matriiseja, jotka kuitenkin suurimmaksi osaksi tulevat vasta toisella matematiikan kurssilla. Yhteistä aikaa MATLABin alkeiden suoritusajalla ja toisella matematiikan peruskurssilla oli vain kaksi viikkoa, mikä voi olla monelle opiskelijalle liian tiukkaa muiden opintojen ohella. Klusterianalyysin pohjalta voidaan kuitenkin todeta, että suurella osalla opiskelijoista suoritusaikaa on ollut riittävästi ja tehtävät eivät ole olleet ainakaan aikaan verraten liian haasteellisia.

Kaiken kaikkiaan MATLABin alkeiden toteutuksen voi katsoa onnistuneen hyvin. Oli ainoastaan joitakin kymmeniä opiskelijoita, jotka olisivat muiden suoritusvaatimusten perusteella saaneet toisesta matematiikan kurssista suoritusmerkinnän mutta merkintä jäi saamatta puutteellisen MATLABin alkeiden suorituksen takia. Nämä opiskelijat voi kuitenkin luokitella yksittäistapauksiksi kaikkien noin 700 ensimmäisen vuosikurssin opiskelijan joukossa. Lisäksi suorittaneiden lukumäärä on suunnilleen suhteellisesti linjassa edellisen vuoden perinteisemmän opetusmateriaalin mukaisen MATLABin alkeiden kokonaisuuden suorittaneiden kanssa. Yhtä tavoiteltua taitoa eli MATLABin käyttötaitojen kehittymistä ei kuitenkaan tällä datalla pystytä arvioimaan. Toisaalta voidaan sanoa, että tällä mallilla on aikaisempia vuosia paremmat edellytykset itsenäiselle oppimiselle, koska jokaisen tulee palauttaa itse tekemänsä tehtävät. Mitään varmaa siitä ei kuitenkaan pystytä sanomaan, koska osan STACK-tehtävistä voi ratkaista niin etteivät MATLAB-käyttötaidot välttämättä kehity.

Parannettavaa nykyisissä MATLABin alkeissa toki on. Tämä on kuitenkin hyvä alusta lähteä kehittämään TTY:n MATLAB-opetusta entisestään. Perusteellisella ja johdonmukaisella tehtäväsuunnittelulla voi STACK-järjestelmällä luoda mutkikkaitakin MATLABiin liittyviä tehtäviä. Tätä järjestelmää voi siis pyrkiä hyödyntämään myös muissa MATLAB-oppimateriaaleissa tai miksei muihinkin laskentaohjelmistoihin keskittyvissä materiaaleissa. Tällä pohjalla voidaan todeta, että MATLAB-oppimateriaalin voi hyvin painottaa koostumaan e-oppimateriaalista, jonka rinnalla voi halutessaan tarjota kontaktiopetusta.

LÄHTEET

[1] L. Ilomäki - Laatua e-oppimateriaaleihin, e-oppimateriaalit opetuksessa ja oppimisessa, OPH 2012. Saatavilla: http://www.oph.fi/julkaisut/2012/laatua_e_oppimateriaaleihin

[2] E. Löfström, K. Kanerva, L. Tuuttila, A. Lehtinen ja A. Nevgi - Laadukkaasti verkossa - Verkko-opetuksen käsikirja yliopisto-opettajalle, Helsingin yliopisto 2010. Saatavilla: http://hdl.handle.net/10138/23899

[3] The MathWorks Inc. - MATLAB, The Language of Technical Computing, vierailtu joulukuussa 2015. Saatavilla: http://se.mathworks.com/products/MATLAB/?s_tid=hp_fp_ml

[4] The MathWorks Inc. - MATLAB Tutorials, vierailtu joulukuussa 2015. Saatavilla: http://se.mathworks.com/videos/getting-started-with-MATLAB-68985.html?s_cid=learn_vid

[5] The MathWorks Inc. - MATLAB Newsgroup, vierailtu joulukuussa 2015. Saatavilla: http://www.mathworks.com/MATLABcentral/newsreader/view_thread/301419

[6] The MathWorks Inc. - MATLAB Academy, vierailtu joulukuussa 2015. Saatavilla: https://MATLABacademy.mathworks.com/R2015b/

[7] The MathWorks Inc. - MATLAB Fundamentals, vierailtu joulukuussa 2015. Saatavilla: https://se.mathworks.com/training-schedule/MATLAB-fundamentals

[8] The MathWorks Inc. - MATLAB Documentation, vierailtu joulukuussa 2015. Saatavilla: http://se.mathworks.com/help/MATLAB/getting-started-with-MATLAB.html

[9] The MathWorks Inc. - MATLAB Videos and webinars, vierailtu joulukuussa 2015. Saatavilla: http://se.mathworks.com/videos/index.html

[10] Tartu Ülikool, Introduction to MATLAB mooc, vierailtu heinäkuussa 2015. Saatavilla: https://moodle.ut.ee/course/view.php?id=3233

[11] Massachusetts Institute of Technology, MITopencourseware: Introduction to MATLAB, vierailtu heinäkuussa 2015. Saatavilla: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-MATLAB-january-iap-2010/index.htm

[12] Massachusetts Institute of Technology - MITopencourseware: Introduction to MATLAB, vierailtu heinäkuussa 2015. Saatavilla: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-MATLAB-january-iap-2010/lecture-notes/

[13] Massachusetts Institute of Technology - MITopencourseware: Introduction to programming in MATLAB, vierailtu heinäkuussa 2015. Saatavilla: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-MATLAB-january-iap-2010/lecture-notes/MIT6_094IAP10_lec01.pdf

[14] Massachusetts Institute of Technology - MITopencourseware: Introduction to programming in MATLAB, vierailtu heinäkuussa 2015. Saatavilla: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-MATLAB-january-iap-2010/lecture-notes/MIT6_094IAP10_lec03.pdf

[15] Massachusetts Institute of Technology - MITopencourseware: Essential Numerical Methods, vierailtu joulukuussa 2015. Saatavilla: http://ocw.mit.edu/courses/nuclear-engineering/22-15-essential-numerical-methods-fall-2014/index.htm

[16] Massachusetts Institute of Technology - MITopencourseware: Essential Numerical Methods Tutorial videos, vierailtu joulukuussa 2015. Saatavilla: http://ocw.mit.edu/courses/nuclear-engineering/22-15-essential-numerical-methods-fall-2014/tutorial-videos

[17] C. Sangwin - Computer Aided Assessment of Mathematics, Oxford University Press 2013. Saatavilla: http://tut.eblib.com/patron/FullRecord.aspx?p=1173592

[18] S. Ingle, V. Duckworth - Enhancing Learning Through Technology In Lifelong Learning : Fresh Ideas, McGraw-Hill Education 2013. Saatavilla: http://tut.eblib.com/patron/FullRecord.aspx?p=1142863

[19] V. Korhonen - Verkko-opetus ja yliopistopedagogiikka, Tampere University Press. Saatavilla: http://tampub.uta.fi/bitstream/handle/10024/68028/verkko_opetus_ja_yliopistopedagogiikka_2004.pdf?sequence=3

[20] Department of Mathematics, University of Utah - MATLAB tutorial, vierailtu joulukuussa 2015. Saatavilla: http://www.math.utah.edu/lab/ms/MATLAB/MATLAB.html

[21] K. Black, Department of Mathematics, University of Georgia - MATLAB Tutorial, vierailtu joulukuussa 2015. Saatavilla: `http://www.cyclismo.org/tutorial/MATLAB/`

[22] Vanderbilt University - Introduction to Programming with MATLAB, vierailtu joulukuussa 2015. Saatavilla: `https://www.coursera.org/course/MATLAB`

[23] D. Eyre - MATLAB Basics and a little beyond, vierailtu joulukuussa 2015. Saatavilla: `http://www.math.utah.edu/~eyre/computing/MATLAB-intro/`

[24] W-H. Steeb - Matrix Calculus and Kronecker Product with Applications and C++ Programs, World Scientific Publishing 1997.

[25] JSXGraph, vierailtu joulukuussa 2015. Saatavilla: `http://jsxgraph.uni-bayreuth.de`

[26] K. Hornik - R FAQ, vierailtu joulukuussa 2015. Saatavilla: `https://CRAN.R-project.org/doc/FAQ/R-FAQ.html`

[27] P. Dalgaard - Introductory Statistics with R, Springer-Verlag New York, Inc. 2002.

[28] The Johns Hopkins University - R Programming, vierailtu joulukuussa 2015. Saatavilla: `https://www.coursera.org/course/rprog`

[29] J. W. Eaton - GNU Octave, vierailtu joulukuussa 2015. Saatavilla: `https://www.gnu.org/software/octave/`

[30] J. W. Eaton - GNU Octave table of contents, vierailtu joulukuussa 2015. Saatavilla: `https://www.gnu.org/software/octave/doc/interpreter/index.html#SEC_Contents`

[31] A. J. Izenman - Modern Multivariate Statistical Techniques, Regression, Classification, and Manifold Learning, Springer 2008.

[32] S. Pohjolainen - Johdatus funktionaalianalyysiin, TTY 2013.

LIITE 1: STACK-TEHTÄVÄ 1

Kysymys 1

Kesken

Kokonaispisteistä 2,00

Tehtävä 1: Peruskomennot

Tidy question | Suorita testitapaukset...

Pisteytys: Jokaisesta kohdasta (a,b,c,d) 0.5 pistettä, koko tehtävästä yhteensä 2 pistettä. Kustakin alakohdasta saa joko 0.5 tai 0 pistettä. Välimuotoja ei ole. Ilmoita vastauskenttiin tarkalleen se muoto, minkä Matlab tulostaa. Älä esimerkiksi lisää kokonaisluvun perään desimaalipistettä ja nollia, koska tehtävä tulkitsee sen tässä tapauksessa virheelliseksi muodoksi.

Tutustu ohjeessa mainittuihin laskutoimituksiin Matlabissa. Kaikki laskukomennot saa näkyviin kirjoittamalla komentoikkunaan *help ++* . Esimerkiksi matriisien kertolaskun notaatio löytyy kohdasta *mtimes* . Kopioi Matlabiin seuraavat rivit:

A=[4,5,6;67,11,1000;1337,101,77];

b=[17; 5; 34];

Kirjoittamalla komentoikkunaan edelliset ilman rivien lopussa olevia puolipisteitä huomaat luoneesi Matlabiin 3×3-matriisin *A* ja pystyvektorin *b*.

Laske Matlabilla seuraavat laskutoimitukset ja kirjoita vastauksesi valmiisiin ruutuihin:

a) A*b =

b) A.^2 =

c) Kopioi Matlabiin seuraavat rivit ja katso mitä ne kirjoittamalla tulostuu:

A(4)

A(3,:)

Laske Matlabilla: A(4)*A(3,:)*b =

d) (Arvostelu: 0.5 pistettä, jos molemmissa vastauslaatikoissa oikea vastaus ja 0 pistettä, jos toisessa tai molemmissa on virhe.) Laske Matlabilla seuraava Kroneckerin tulo:

C=kron(ones(3,2),(1:10)')

Mitkä ovat matriisin C korkeus ja leveys? (Kirjoita komentoikkunaan *help kron*, jos haluat tietää Kroneckerin tulon laskukaavasta enemmän.)

korkeus:

leveys:

LIITE 2: STACK-TEHTÄVÄ 2

Kysymys 1

Kesken

Kokonaispisteistä 2,00

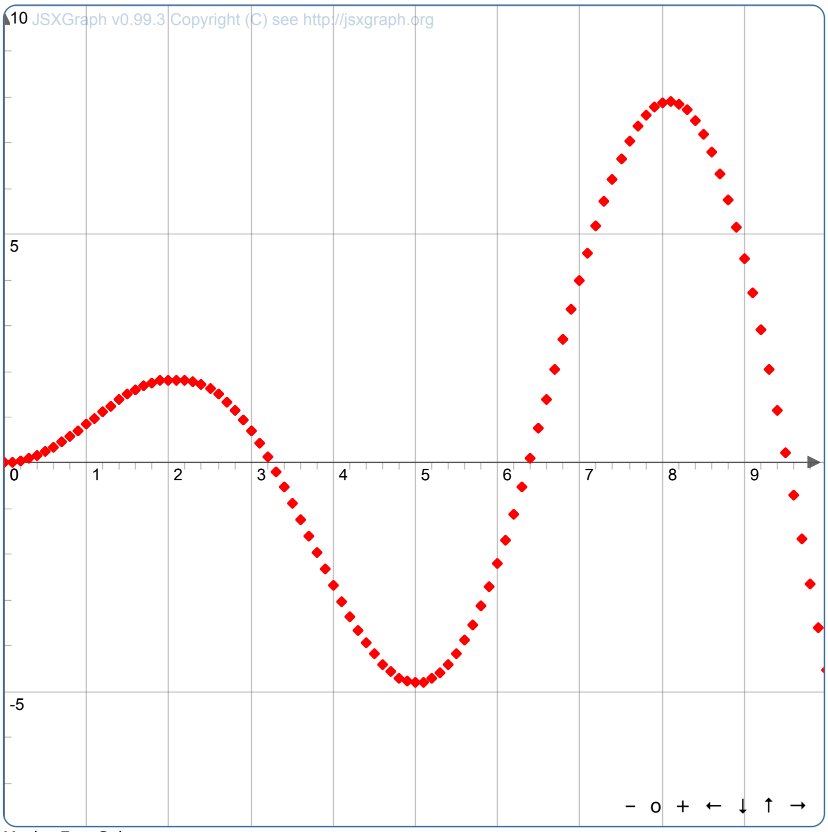
Tehtävä 2: Kuvaajien piirtäminen ja käsittely Tidy question | Suorita testitapaukset...

Pistetytys: a)- ja b)-kohdista molemmista 0.5 pistettä, c)-kohdasta 1 piste. Sekä a)- b)- että c)-kohdassa tehtävän tulee olla kokonaan oikein, jotta pisteet voi saada. Jos c-kohdassa yrittää vilppiä, niin Matlab-harjoitusosioon saa -3 pistettä, eli koko suoritus hylätään suoraan. Vajaita pisteitä ei anneta.

a) Piirrä Matlabin plot-komennolla funktion xsin(x) kuvaaja siten että sen koordinaattipisteet ovat kokopunaisia timantteja ja niitä on sata kappaletta tasaisin välein reaalilukuvälillä $x \in [0, 10]$. Mallikuva tavoiteltavasta kuvaajasta on alla. Hae kuvaajan ominaisuudet get-komennolla ja kirjoita kohdassa 'MarkerFaceColor' lukevan vektorin alkiot. Funktion piirtämiseen tarvittavat komennot ovat alla. %-merkkien jälkeen tulevaa saman rivin tekstiä ei tarvitse kirjoittaa, sillä se on kommentti, jonka tarkoitus on selventää, mitä rivi kirjoittamalla tapahtuu.

```
x=0:0.1:10; h=plot(x,x.*sin(x),'rd','MarkerFaceColor','r');
hold on; %säilyttää kuvan jatkossa
grid on; grid minor; %lisää ruudukon kuvaajaan
get(h)
```

LIITE 2: STACK-TEHTÄVÄ 2



MarkerFaceColor:

b) Skaalaa edellisen kuvaajan koordinaattiakselit uudelleen kirjoittamalla komento axis([5.1564, 5.7564, -5.6564, -1.8941]); (HUOM! Jos kopioit komennon tästä, niin Matlab ei välttämättä tunnista miinusmerkkejä. Ongelmasta selviää, kun kirjoittaa miinusmerkit uudelleen komentoikkunaan.)

Kuinka monta pistettä uudessa kuvaajassa näkyy?

Pisteiden lukumäärä:

c) Aseta saman kuvaajan koordinaattiakselit takaisin ennalleen komennolla

axis([0,10,-8,10]);

Luo muuttuja op, jonka arvo on oma opiskelijanumerosi. Sen jälkeen tulosta kuvaajastasi pdf ja palauta tiedosto Moodleen erillisessä palautuskohdassa. Vastaa alla olevaan valintakysymykseen rehellisesti. Tätä varten kirjoita seuraavat komennot:

```
op=xxxxxx; (x-kirjaimien tilalle tulee opiskelijanumero)
title(['Opiskelijanumeroni on ' num2str(op)])
print(gcf,'-dpdf','kuvatesti.pdf')
```

LIITE 2: STACK-TEHTÄVÄ 2

Sain palautettua omalla opiskelijanumerollani varustetun pdf:n. Vastaa rehellisesti sen mukaan, saitko palautettua vai et. Jos jää kiinni vilpistä, niin Matlab-harjoituspaketin suoritus hylätään.

(tosi/true: 1 piste, epätosi/false: 0 pistettä) Ei vastattu

Lukitsen vastaukseni

Aloita uudelleen | Tallenna | Täytä oikeat vastukset | Palauta ja lopeta | Sulje esikatselu

Teknistä tietoa

Käytössä oleva käyttäytyminen: Adaptive mode (multi-part questions)

Minimum fraction: 0

Maximum fraction: 1

Question variant: 99

Kysymyksen yhteenveto: \({5.1564}\),\({5.7564}\),\({-5.6564}\),\({-1.8941}\),\({6}\)

Oikean vastauksen yhteenveto: ans1: matrix([1,0,0]) [score]; ans2: 6 [score]; ans3: true [score]

Response summary:

Kysymyksen tila: todo

Suorituskerran asetukset

Näyttöasetukset

LIITE 3: STACK-TEHTÄVÄ 3

Kysymys 1

Kesken

Kokonaispisteistä 2,00

Tehtävä 3: Funktiot

Tidy question | Suorita testitapaukset...

Pistetytys: a)-kohdasta 1 piste, b)-kohdasta yhteensä 1 piste siten että kolmen ensimmäisen tyhjän laatikon oikein saamisesta 0.5 pistettä ja vastaavasti kolmen jäkimmäisen vastauslaatikon oikein saamisesta 0.5 pistettä. Muista käyttää desimaaliluvuissa desimaalipistettä eikä -pilkkua! Ilmoita desimaaliluvut Matlabin tulostustarkkuuden mukaisesti (neljä desimaalia) ja muuta tarkkuus tarvittaessa oikeaksi kirjoittamalla komentoikkunaan *format short* .

a) Lataa koneellesi Moodlesta tiedosto testiptiedosto.p ja aja se Matlabissa seuraavalla argumentilla: testiptiedosto(2.28)

Minkä lukuarvon Matlab antaa vastaukseksi?

b) Muodosta Matlabilla funktio, joka laskee toisen asteen polynomin $px = ax^2 + bx + c$ reaalisten juurien lukumäärän diskriminantin avulla ja ilmoittaa sen näytöllä. Avaa editoriin tiedosto juurtenlkm.m ja luo funktio sinne. Mitä tulee kirjoittaa seuraaviin tyhjiin laatikoihin, jotta funktio toimisi oikein? Funktion argumentti p on kolmen alkion vaakavektori, $p = [a, b, c]$. Käytä vektorin alkioista järjestysmerkintää, eli $p(1)$, $p(2)$ ja $p(3)$.

function diskriminantti = juurtenlkm(p)

diskriminantti = p( )^2-4*p( )*p( ); (0.5 pistettä.)

if diskriminantti >

disp('Polynomilla on kaksi reaalista juurta.')

elseif diskriminantti ==

disp('Polynomilla on yksi reaalinen juuri.')

elseif diskriminantti < (0.5 pistettä)

disp('Polynomilla ei ole reaalisia juuria.')
else
disp(['Jotain meni pieleen. D = ', num2str(diskriminantti)])
end

Lukitsen vastaukseni

Aloita uudelleen | Tallenna | Täytä oikeat vastukset | Palauta ja lopeta | Sulje esikatselu

Teknistä tietoa

Käytössä oleva käyttäytyminen: Adaptive mode (multi-part questions)

Minimum fraction: 0

Maximum fraction: 1

Question variant: 86

Kysymyksen yhteenveto: \({2.28}\)

LIITE 4: STACK-TEHTÄVÄ 4

Kysymys 1

Kesken

Kokonaispisteistä 2,00

Tehtävä 4: Pienimmän neliösumman menetelmä ja sovitteet

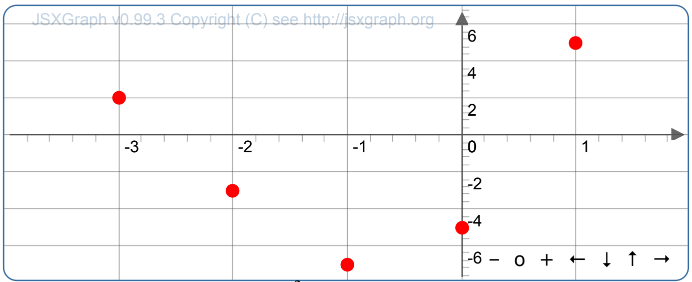
Tidy question | Suorita testitapaukset...

Pistetytys: Täysin oikeasta a)- ja b)-kohdasta molemmista 1 piste. Ilmoita tulokset Matlabin tulostustarkkuuden mukaisesti. Voit jälleen varmuuden vuoksi kirjoittaa komentoikkunaan ennen tehtävän ratkaisua *format short* .

Tässä tehtävässä on tarkoitus tehdä annettuun dataan toisen asteen pienimmän neliösumman polynomisovitus. Annetut pisteet on piirretty kuvaajaan punaisella ja niiden koordinaatit ovat

x=[-3; -2; -1; 0; 1];

y = [2; -3; -7; -5; 5];



Toisen asteen polynomin $p(x) = ax^2 + bx + c$ sovitus annettuun dataan voidaan kirjoittaa lineaariseksi yhtälöryhmäksi (matriisimuotoon), missä jokainen piste muodostaa yhden yhtälön

$$\mathrm{A} * \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ -7 \\ -5 \\ 5 \end{bmatrix}$$

a) Anna matriisin A alkiot. Jos vastaat oikein, saat matriisin sellaisessa muodossa, jonka voit kopioida suoraan Matlab-ohjelmaan.

A =

b) Ratkaise Matlabilla toisen asteen polynomin kertoimet a, b ja c sekä "kenoviivalla" komennolla $\mathrm{A} \backslash \mathrm{y}$, että Matlabin *polyfit(x,y,2)* komennolla. Syötä kertoimien arvot oheisiin kenttiin Matlabin tulostustarkkuuden mukaisesti.

A\y =

LIITE 4: STACK-TEHTÄVÄ 4

polyfit(x,y,2) =

Lukitsen vastaukseni

Aloita uudelleen | Tallenna | Täytä oikeat vastukset | Palauta ja lopeta | Sulje esikatselu

Teknistä tietoa

Käytössä oleva käyttäytyminen: Adaptive mode (multi-part questions)

Minimum fraction: 0

Maximum fraction: 1

Question variant: 36

Kysymyksen yhteenveto: \({\left[2 , -3 , -7 , -5 , 5 \right]}\)

Oikean vastauksen yhteenveto: ans1: matrix([9,-3,1],[4,-2,1],[1,-1,1],[0,0,1],[1,1,1]) [score]; ans2: matrix([2.5714],[5.5429],[-3.7714]) [score]; ans3: matrix([2.5714,5.5429,-3.7714]) [score]

Response summary:

Kysymyksen tila: todo

Suorituskerran asetukset

Näyttöasetukset

LIITE 5: STACK-TEHTÄVÄ 5

Kysymys 1

Kesken

Kokonaispisteistä 2,00

Tehtävä 5: Tiedoston lataaminen ja käsittely Matlabissa Tidy question | Suorita testitapaukset...

Pisteytys: a)- ja b)-kohdista molemmista 1 piste. Jälleen kerran ilmoita vastaus Matlabin tulostustarkkuudella. *format short*

a) Lataa Moodlesta exceltiedosto.xlsx koneellesi ja lue se Matlabilla muuttujaan A joko "import data" -työkalun avulla tai seuraavilla komennoilla:

```
filename='exceltiedosto.xlsx';
A=xlsread(filename);
```

Ilmoita matriisin A seuraava arvo:

A(20)=

b) Laske seuraava determinantti ja ilmoita se Matlabin tulostustarkkuudella:

det(2*A(5:2:15,5:2:(5+10)))=

Lukitsen vastaukseni

Aloita uudelleen | Tallenna | Täytä oikeat vastukset | Palauta ja lopeta | Sulje esikatselu

Teknistä tietoa

Käytössä oleva käyttäytyminen: Adaptive mode (multi-part questions)

Minimum fraction: 0

Maximum fraction: 1

Question variant: 67

Kysymyksen yhteenveto: \({20}\)

Oikean vastauksen yhteenveto: ans1: 0.0277 [score]; ans2: -0.67396 [score]

Response summary:

Kysymyksen tila: todo

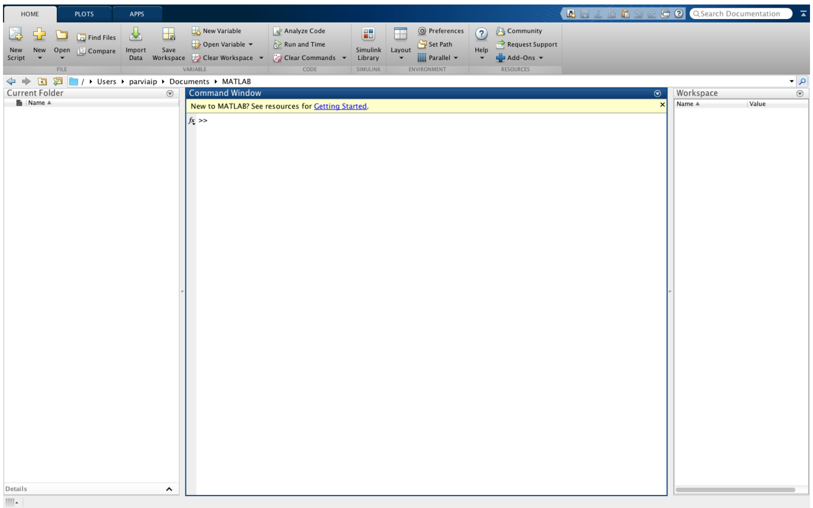
Suorituskerran asetukset

Näyttöasetukset

LIITE 6: TEHTÄVÄN 1 OHJEPDF

TEHTÄVÄ 1: PERUSKOMENNOT

Tämän tehtävän tarkoituksena on johdattaa opiskelija Matlabin käyttöön siten että liikkeelle lähdetään laskentaohjelmiston käytön perusteista ja jatketaan matriisilaskennan komentoja harjoittelemalla. Kun Matlabin avaa, ruudulle tulee näkyviin kuvan 1 kaltainen näkymä.



Kuva 1: Käyttäjälle aukeava Matlabin päänäkymä. Kolme ikkunaa vasemmalta oikealle ovat kansio (current folder), komentoikkuna (command window) ja työtila (workspace).

Current folder merkitsee aktiivisena olevaa kansiota. Esimerkiksi Matlabilla luodut ohjelmat tallentuvat siihen kansioon, joka current folder -valikossa on aktiivisena. Tämä kannattaa huomioida myös silloin kun Matlabiin lataa tiedostoja komentoikkunasta, koska ohjelma lukee tietoja tämän valikon kansiosta. Workspaceen eli työtilaan tallentuvat kaikki alustetut muuttujat arvoineen ja tyyppeineen. Komentoikkunaan (command window) puolestaan kirjoitetaan komentoja kuten laskutoimituksia. Muuttujia alustetaan Matlabissa yhtäsuuruusmerkillä. Kirjoittamalla komentoikkunaan esimerkiksi

a=2

alustaa Matlabiin muuttujan a, jonka arvoksi tulee reaaliluku 2. Alustuksen jälkeen huomaat, että muuttuja nimeltä a tulee lukuarvoineen näkyville myös työtilaan. Jos komennon perään lisää puolipisteen ; niin kirjoitettu komento tai laskutoimitus ei tulostu komentoikkunaan näkyville mutta tallentuu Matlabin muistiin kuitenkin. Vektorit ja matriisit kirjoitetaan Matlabiin hakasulkeilla. Esimerkiksi komento:

LIITE 6: TEHTÄVÄN 1 OHJEPDF

b=[2,4,9]

luo Matlabiin 1×3 -vaakavektorin b. Matlab alustaa vektorit oletusarvoisesti vaakavektoreiksi. Alkioiden välissä voi olla pilkku tai välilyönti. Huomioi, että desimaaliluvuissa on käytössä niin Matlabissa kuin useimmissa muissakin ohjelmistoissa desimaalipiste eikä -pilkku. Matriiseissa syntaksi on sama kuin vektoreissa ja vaakarivin vaihtokohtaan tulee puolipiste, kuten huomataan esimerkiksi seuraavassa 3×3 -matriisissa:

A=[2,4,9;5,8,11;27,65,101]

Matriisien konjugaattitranspoosi muodostetaan Matlabissa kirjoittamalla matriisin perään heittomerkki '. Jos matriisissa on vain reaalisia alkioita, niin konjugaattitranspoosimerkintää voi luonnollisesti käyttää samaan tapaan kuin tavallista transpoosiakin. Tavallisen transpoosin saa lisäämällä heittomerkin eteen pisteen eli .'.

Peruslaskutoimituksissa käytetään seuraavanlaisia merkkejä: + yhteenlasku, - vähennyslasku, * kertolasku, / jakolasku ja ˆ potenssiin korotus. Samat syntaksit pätevät myös monissa muissakin laskentaohjelmistoissa, joten nämä merkinnät kannattaa painaa mieleen. Matriisimuotoisten muuttujien tapauksessa laskutoimitukset ovat hieman poikkeavia sklaareilla laskemiseen verrattuna. Esimerkiksi kertolasku * tarkoittaa määritelmän mukaista matriisituloa. Jos kertolaskumerkin eteen lisää pisteen .*, niin Matlab laskee laskun alkioittain. Esimerkiksi edellä määritetyllä vektorilla b pisteellinen laskutoimitus lasketaan näin: b.*b=[2*2,4*4,9*9]=[4,16,81]. Lisätietoa peruslaskutoimituksista löytää kirjoittamalla komentoikkunaan *help* ++. Ikkunaan tulostuu listaus eri aritmeettisista laskutoimituksista, joita klikkaamalla tulostuu lisätietoa aiheesta. Esimerkiksi matriisitulon nimi listassa on *mtimes*. *help* on muutenkin käyttökelpoinen komento, jonka avulla saa lisätietoa eri komentojen vaatimista parametreista. Komento toimii niin, että *help* -sanan perään kirjoittaa sen komennon nimen, josta haluaa tietää enemmän. Toinen vaihtoehto on kirjoittaa *help* yksinään, jolloin voi suunnistaa hakemistopolkuja pitkin tietoa etsien.

Matlabissa voi muodostaa pienemmistä matriisimuotoisista muuttujista suurempia konkatenoimalla. Tämä tapahtuu normaalisti vektorinotaatiolla ja pilkuilla tai puolipisteillä. Käytetään esimerkissä samoja edellä määriteltyjä vektoreita ja matriiseja. Komento c=[A,b'] luo Matlabissa matriisin c, jossa matriisi A ja vektorin b konjugaattitranspoosi b' on konkatenoitu pystysuunnassa vierekkäin, eli vektorin b' alkiot ovat matriisissa c samassa järjestyksessä matriisin A oikealla puolella. Vastaavasti puolipisteellinen notaatio d=[A;b] luo matriisin d, jossa matriisi ja vektori on konkatenoitu vaakasuunnassa allekkain, eli vektorin b alkiot ovat matriisin A alapuolella. Tässäkin asiassa pitää muistaa matriisien dimensiot kuten peruslaskutoimituksissakin. Matriiseja ei voi konkatenoida, jos matriisien vaaka- tai pystyrivit eivät ole keskenään samassa dimensiossa.

Matlabissa voi kätevästi muodostaa matriiseja, joiden kaikki alkiot ovat nollia tai ykkösiä. Näitä tehdään komennoilla $zeros()$ tai $ones()$ vastaavasti, ja sulkuihin kirjoitetaan matriisinotaation tapaan vaaka- ja pystyrivien lukumäärä pilkulla erotettuna. Esimerkiksi komento zeros(5,2) muodostaa 5×2-matriisin, jonka kaikki alkiot ovat nollia. Komennolla eye(n) muodostetaan $n \times n$-identiteettimatriisi. Matriisimuotoisten muuttujien koon saa kätevästi selville kirjoittamalla komentoikkunaan size(). Sulkujen sisään tulee matriisimuuttujan nimi. Lopputuloksessa näkyy ensin vaakarivien lukumäärä ja sen jälkeen pystyrivien lukumäärä. Kokeile esimerkiksi kirjoittamalla size(b).

Matriiseista voi myös hakea järjestyksessä tietyn/tiettyjä alkioita. Jatketaan aiemmin tässä tekstissä määritellyllä matriisilla A. Komennolla A(5) Matlab hakee matriisista A järjestyksessä viidennen alkion. Alkioiden järjestys on tässä tapauksessa pystyriveittäin ylhäältä alas ja vasemmalta oikealle. $A(i, j)$-notaatiolla voi puolestaan hakea matriisin A alkion, joka sijaitsee i:nnellä vaakarivillä ja j:nnellä pystyrivillä. Kaksoispisteellä : Matlab tulostaa ruudulle matriisista vaaka- tai pystyrivin kokonaisuudessaan. Esimerkiksi komento A(3,:) tulostaa matriisin A kolmannen vaakarivin. Vastaavasti komento A(:,2) tulostaa toisen pystyrivin. Kokeile näitä merkintöjä, niin ne tulevat tutuiksi.

LIITE 7: TEHTÄVÄN 2 OHJEPDF

TEHTÄVÄ 2: KUVAAJIEN PIIRTÄMINEN JA KÄSITTELY

Tässä tehtävässä harjoitellaan kuvaajien piirtämistä ja käsittelemistä Matlabilla. Matlabilla on mahdollista siististi piirtää niin kaksi- kuin kolmiulotteisiakin kuvia ja myös erilaisten sovitteiden tekeminen luonnistuu hyvin, kuten neljännessä harjoituksessa huomataan. Kuvaajan voi piirtää *plot*-komennolla. Esimerkiksi seuraavalla komentoparilla voi piirtää sinifunktion kuvaajan:

```
x=linspace(1,10);
h=plot(x,sin(x))
```

Plot-komento tarvitsee siis kaksi argumenttia, jotka kuvaavat x- ja y-akselien koordinaatteja. *Linspace* on käyttökelpoinen tapa muodostaa tasaisin välein mittapisteitä. *Linspace* luo vektorin, jossa on sata pistettä tasaisin välein suluissa annettujen argumenttien välillä. Koordinaattipisteet voivat myös muodostua mistä tahansa kahdesta samankokoisesta vektorista ja niitä voi piirtää samaan kuvaajaan useampiakin kerralla. Esimerkiksi komento *plot(x1,y1,x2,y2)* piirtää samaan kuvaan käyrän $y1$ $x1$:n suhteen ja käyrän $y2$ $x2$:n suhteen. Kuten ehkä olet jo huomannut, kuvaaja tulostuu erilliseen ikkunaan, jonka nimi on *figure*. Kaikki uudet plotilla tehdyt muutokset kirjautuvat päälle aktiiviseen figureen. Jos haluaa säilyttää edellisen kuvaajan, niin uuden *figure*-ikkunan saa luotua kirjoittamalla komentoikkunaan yksinkertaisesti *figure* .

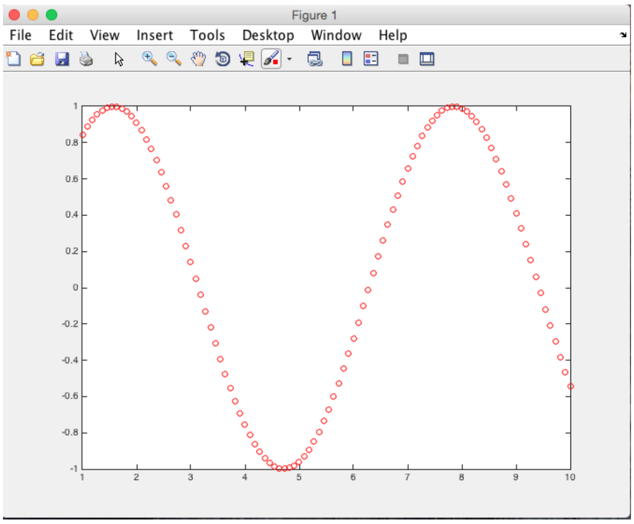
Plot yhdistää koordinaattipisteet oletusarvoisesti sinisillä suorilla viivoilla kuin pisteestä pisteeseen -tehtävissä. Tämä ominaisuus ei ole suotavaa esimerkiksi mittausdatan analysoinnissa, joten voi olla järkevämpää piirtää pisteet erillisinä kuvaajaan. Sen voi toteuttaa lisäämällä kolmannen merkkijonoargumentin *plot*-komennon argumenttien perään. Kokeile kirjoittaa Matlabiin seuraava rivi:

```
h=plot(x,sin(x),'ro')
```

Komento luo kuvan 1 kaltaisen sinikäyrän. Punainen väri on peräisin kolmannen argumentin kirjaimesta r ja rengasmuotoinen mittapiste tulee merkistä o. Tässä ja useimmissa muissakin merkkijonoja (ei-numeroita) käsittelevissä tilanteissa merkkijonon ympärillä pitää olla heittomerkit '', jotta komento ei antaisi virhettä. Kirjoittamalla *help plot* saa selville kaikki mahdollisuudet, mitä merkkijonoargumenttiin voi kirjoittaa. Esimerkeistä huomaa, että kuvaaja on alustettu muuttujaan h. Kuvaajan saa myös piirrettyä alustamatta sitä johonkin muuttujaan, mutta muuttuja-alustuksella saman kuvaajan asetuksia voi muuttaa kätevästi. Kirjoittamalla *get(h)* saa haettua ja tulostettua kuvaajan h ominaisuudet komentoikkunaan. Vastaavasti komennolla set(h) niitä voi asettaa toisenlaisiksi. Sulkujen sisälle tulee syntaksina (h,'ominaisuus','uusi arvo'). Hae kuvaajan h ominaisuudet

LIITE 7: TEHTÄVÄN 2 OHJEPDF

get-käskyllä. Muuta sen jälkeen seuraavalla komennolla kuvaajan *h* datapisteet punaisiksi plusmerkeiksi:



Kuva 1: Plotilla piirretty kuvaaja aukeaa tämänkaltaiseen erilliseen figure-ikkunaan.

set(h,'Marker','+')

Seuraavassa muutamia käyttökelpoisia matlabkomentoja kuvaajien piirtämiseen ja käsittelyyn liittyen:

figure

Luo tyhjän kuvaajapohjan. Käyttökelpoinen esimerkiksi silloin jos tarvitsee tehdä useita erillisiä kuvaajia, koska oletusarvoisesti uuden kuvaajan piirtäminen aina korvaa edellisen.

axis

Tällä komennolla voi säädellä koordinaattiakselien skaalausta.

box

Säätelee kuvaajaikkunan laatikon asetuksia. Esimerkiksi *box on* laatikoi aktiivisen kuvaajan ja *box off* kytkee laatikon pois päältä.

grid

Säätelee ruudukon asetuksia vastaavasti kuin *box*-komentokin.

LIITE 7: TEHTÄVÄN 2 OHJEPDF

hold on

Muuttaa viimeksi luodun kuvaajan aktiiviseksi eli komento mahdollistaa kyseisen kuvaajan ominaisuuksien muuttelemisen. Muussa tapauksessa esimerkiksi uusi *plot*-komento korvaisi edellisen.

subplot

Mahdollistaa usean kuvaajan piirtämisen samaan *figure*-ikkunaan.

print

Tulostaa kuvaajan erilliseen tiedostoon.

Plot ei ole ainut Matlabin komento, jolla kuvaajia voi piirtää. Esimerkiksi *polar*-komennolla voi piirtää napakoordinaattikuvaajia. Kolmiulotteisia kuvaajia voi piirtää esimerkiksi *plot3*- tai *surf*-komennoilla. Viimeksi mainittu muodostaa värillisen pintakuvaajan.

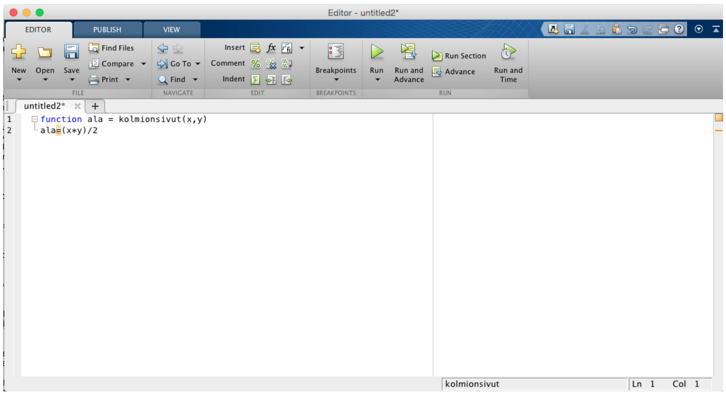
LIITE 8: TEHTÄVÄN 3 OHJEPDF

TEHTÄVÄ 3: FUNKTIOT

Matlabilla voi koodata omia ohjelmiaan, kuten funktioita, joiden tekemistä harjoitellaan tämän tehtävän kautta. Funktiot ovat objekteja, jotka laskevat käyttäjän antamista syötteistä jonkun toisen arvon. Esimerkiksi trigonometriset funktiot $\sin(x)$ ja $\cos(x)$ ovat valmiita Matlabin funktioita. Funktioiden syötteet voivat olla muun muassa yksittäisiä lukuarvoja tai vektoreita.

Matlabissa itse luodut ohjelmat tallenetaan omiin tiedostoihinsa. Nämä tiedostot ovat muun muassa .m-päätteisiä tai .p-päätteisiä tiedostoja. Useimmiten ajettavat tiedostot ovat .m-tiedostoja. Tässä tehtävässä esimerkkinä ajettavaksi tiedostoksi annetaan .p-tiedosto. Tämä tiedostotyyppi on muuten samanlainen kuin .m-tiedosto mutta sen sisältö on käyttäjältä salattu. Funktioiden sisältö kirjoitetaan erillisessä editorissa, joka avataan *edit*-komennolla. Komento toimii niin, että sen perään kirjoitetaan tiedoston nimi, jonka perässä voi olla .m mutta se ei ole välttämätöntä. Editori aukeaa sen jälkeen asetuksista riippuen joko erilliseen ikkunaan tai komentoikkunan yläpuolelle. Kuva editori-ikkunasta näkyy kuvassa 1. Ota huomioon, että mikäli lähdet editoimaan jotain Matlabiin sisään rakennettua funktiota tai sellaista tiedostoa, joka on jo ennestään olemassa, niin editori ei näytä tyhjää. Huomioi myös, että Matlab hakee tiedostoja senhetkisestä aktiivisesta kansiosta (current folder) ja myöskin tallentaa uudet tiedostot juurikin sinne. Tehdään seuraavaksi esimerkkifunktio, joka laskee käyttäjän antamista kolmioiden sivun pituuksista kolmion alan. Luo editoriin tiedosto nimeltä *kolmionsivut.m* seuraavasti:

edit kolmionsivut TAI edit kolmionsivut.m



Kuva 1: Matlabin editori-ikkuna. Editoriin kirjoitetut komennot eivät toteudu editorissa, ja tunnistetut sanat värjäytyvät.

Funktiotiedoston alussa täytyy olla funktiomäärittely. Kirjoita edellä luotuun tiedostoon ensimmäinen rivi:

LIITE 8: TEHTÄVÄN 3 OHJEPDF

```
function ala = kolmionsivut(x,y)
```

Nyt funktion ulostuloarvoksi on määritelty *ala* sekä käyttäjän antamiksi syötteiksi x ja y. Funktio on *kolmionsivut*, joka laskee ulostuloarvon. Funktiomäärittelyn jälkeen kirjoitetaan itse laskutoimitus:

```
function ala = kolmionsivut(x,y)
ala=(x*y)/2
```

Tällaisessa muodossaan funktio toimii jo siinä mielessä riittävällä tasolla, että se laskee kolmion alan, kun kirjoittaa komentoikkunaan esimerkiksi `kolmionsivut(2,6)`, ja tulostaa lopputuloksen. Funktion tuloksen ulkoasusta saa hieman fiksumman esimerkiksi muuttamalla luvun merkkijonoksi komennolla *num2string* ja tulostamalla sen tekstinä (*disp*). Kirjoita seuraavat rivit ohjelmaan lisäksi:

```
function ala = kolmionsivut(x,y)
ala=(x*y)/2;
A=num2str(ala);
X=['Kolmion ala on ', A,'.'];
disp(X)
```

Lisätyt rivit luovat merkkijonon, joka näyttää lopputuloksen sanallisessa muodossa. Ohjelmaa voi kommentoida %-merkeillä. Kaikki kyseisen merkin jälkeen tuleva saman rivin teksti on kommentti eikä vaikuta ohjelman toiminnallisuuteen mitenkään. Kommenteilla voikin esimerkiksi selventää lukijalle joitakin ohjelman toiminnallisuuksia tai sitten testata ohjelman tietyn kohdan toimintaa muuttamatta koodia lopullisesti. Ehtolauseet ovat näppärä tapa luoda ohjelmiin lisätoiminnallisuuksia. Samaan ohjelmaan voidaan luoda vaikkapa sellainen *if*-lause, joka antaa virheilmoituksen silloin kun kolmion alasta tulee negatiivinen. Täydennä funktiota:

LIITE 8: TEHTÄVÄN 3 OHJEPDF

```
%Tämä funktio laskee kolmion pinta-alan.
function ala = kolmionsivut(x,y)
ala=(x*y)/2;
if ala < 0
disp('Kolmion ala ei voi olla negatiivinen.')
else
A=num2str(ala);
X=['Kolmion ala on ', A,'.'];
disp(X)
end
```

Muita mahdollisia silmukoita ovat esimerkiksi *for*-silmukka, jonka sisällä oleva ehto täyttyy niin monta kertaa kuin määritetään, ja *while*-silmukka, toistaa sisältöään niin pitkään kuin jokin ehto on voimassa.

LIITE 9: TEHTÄVÄN 4 OHJEPDF

TEHTÄVÄ 4: PIENIMMÄN NELIÖSUMMAN MENETELMÄ JA SOVITTEET

Pienimmän neliösumman menetelmä (least squares method) on matemaattinen malli, jolla voi määrittää kätevästi pistejoukkoon sovitteita ja sovitteiden avulla voi ennustaa säännönmukaisuutta. Monesti esimerkiksi tutkimuksessa saadut mittapisteet eivät noudata täysin teorian mukaista säännönmukaisuutta, jolloin sovitteiden tekeminen säännönmukaisuuden löytämiseksi auttaa merkittävästi.

Tavallisimmat sovitteet ovat eriasteisia polynomeja mutta myös esimerkiksi logaritmisia ja eksponentiaalisia sovitteita käytetään. Datapisteiden asettumisesta riippuu, millainen sovite on paras mihinkin tilanteeseen. Jos on esimerkiksi tutkimassa jotain luonnonilmiötä, jonka teoriasta tietää, millaista säännönmukaisuutta ilmiön pitäisi noudattaa, niin mittausdataan voi sitten yrittää määrittää samantyyppistä sovitetta. Havainnollistetaan sovitteiden tekemistä Matlabilla muodostamalla seuraava joukko pisteitä kuvaajaan:

```
x=[1,2,3,4,5,6,7,8,9,10];
y=[5.3,6.3,6.88,8.06,8.64,10.12,11.06,12,13.24,14.18];
plot(x,y,'o')
hold on
```

Kuvaajasta näkee, että pisteet asettuvat suunnilleen suoralle, joten hyvä sovite tälle datalle voisi olla suora eli ensimmäisen asteen polynomi. Suoran sovittamista pistejoukkoon sanotaan lineaariseksi regressioksi. Yksi tapa muodostaa sovite on *polyfit*-komento. *Polyfit* tarvitsee argumenteikseen pisteet, joihin sovite tehdään, sekä lisäksi sovitepolynomin asteen. Komento laskee sovitteen kertoimet eli suoran tapauksessa suoran kulmakertoimen sekä suoran ja y-akselin välisen leikkauspisteen. Näiden parametrien avulla voi sitten muodostaa suoran, joka suunnilleen kulkee pistejoukon läpi. Nämä arvot tulostuvat komentoikkunaan vaakavektorina, joista ensimmäinen arvo on kulmakerroin ja toinen leikkauspiste vastaavasti. Muodostetaan sovitesuora:

```
h=polyfit(x,y,1)
plot(h(1)*x+h(2))
```

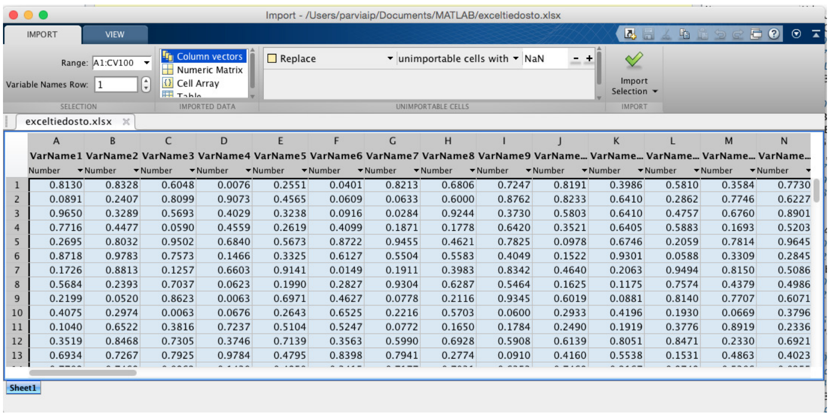
Äskeinen komentosarja muodostaa samaan kuvaajaan pistejoukon lisäksi suoran. Suoran kertoimet voi luonnollisesti kirjoittaa lukuarvoina, eikä tuollaista kerroinnotaatiota tarvitse välttämättä käyttää. Suoran voi tehdä muutakin kautta. Kun avaa *figure*-ikkunan ja valitsee työkalupalkin *tools* -valikosta kohdan *basic fitting* , niin sivuun aukeaa uusi ikkuna, josta voi säädellä sovitteita graafisen käyttöliittymän avulla. Ikkunassa on liukuvalikko, josta voi valita sovitteen tyypin. Tässä tapauksessa tyyppi on *linear*. Jos vielä klikkaa rastin kohtaan *Show equations*, niin huomaa, että kuvaajaan suoran viereen ilmestyy suoran yhtälö, jonka kertoimet ovat lähes samat kuin äsken *polyfit*-komennolla lasketut.

Matlabissa on myös monipuolisempi työkalupaketti sovitteiden tekemiseen. Tämä työkalu on nimeltään *cftool*, ja siihen pääsee käsiksi kirjoittamalla komentoikkunaan *cftool*. Cftoolilla voi valita useita erilaisia sovitteita ja lisäksi voi muun muassa jättää huomiotta joitakin datapisteitä, jos ne ovat selkeästi karkeita virheitä. Kannattaa tutustua hieman cftooliin, koska se voi myöhemmin osoittautua käyttökelpoiseksi työkaluksi.

LIITE 10: TEHTÄVÄN 5 OHJEPDF

TEHTÄVÄ 5: TIEDOSTON LATAAMINEN JA KÄSITTELY MATLABISSA

Matlab osaa myös käsitellä joitakin muista ohjelmista peräisin olevia tiedostoja. Tästä ominaisuudesta on hyötyä siinä mielessä, että Matlabin laskuoperaatioita voi käyttää dataan hyväkseen. Matlabin tukemia tiedostoja ovat muun muassa tekstitiedostot .txt, excel-tiedostot .xls, jotkin kuvatiedostot kuten .jpg, .jpeg, .gif, .png sekä myös jotkin ääni- ja videotiedostot, kuten .mp4. Matlabissa on erillisiä komentoja eri tiedostotyyppien lukemiseen, kuten *load*, *imread* tai *audioread*. Tiedoston lukemisen voi tehdä myös manuaalisesti työkalupalkin *import data* -työkalun avulla. Kun sen valikon avaa, ruudulle ilmestyy hakemisto, josta voi hakea Matlabin tukeman tiedoston. Sen jälkeen erilliseen ikkunaan aukeaa kuvan 1 kaltainen näkymä.



Kuva 1: Ikkunanäkymä data importista. Tiedostotyypistä riippuen tuotava data saattaa näkyä eri tavoin.

Tämän nimenomaisen tehtävän päämääränä on lukea ja käsitellä Excelillä tehtyä .xls-tiedostoa. Operaation voi tehdä *data importin* kautta tai sitten komennolla *xlsread*. Komentoa käytettäessä syntaksi on seuraava:

a=xlsread('exceltiedosto.xls')

Komento tallentaa muuttujaan *a* tiedoston nimeltä exceltiedosto.xls numeeriset arvot matriisina. Huomaat, että matriisimuotoinen muuttuja ilmestyy Matlabin workspaceen. Tiedostopääte ei ole välttämätön luettavassa tiedostossa, mutta heittomerkit tiedostonimen ympärillä täytyy muistaa. Oheisella komennolla voi myös lukea xls-tiedoston tekstit ja raakadatan sekä tallentaa nekin Matlabiin. Komennolla *help xlsread* selviää, miten tämä onnistuu.

LIITE 10: TEHTÄVÄN 5 OHJEPDF

Nyt muuttujaan *a* on tallennettu matriisimuotoinen muuttuja, jonka alkiot koostuvat tiedoston exceltiedosto.xls numeerisista arvoista. Tätä muuttujaa voi käsitellä kuin mitä tahansa muutakin matriisia Matlabissa, ja samat laskukomennot pätevät siihenkin.

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

Table of Contents

Tuodaan dataa MATLABiin taulukosta

```
close all
clear all
```

Tuodaan data MATLABiin

```
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet1');
RAW = raw(2:end,:);
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet2');
RAW = [RAW;raw(2:end,:)];
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet3');
RAW = [RAW;raw(2:end,:)];
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet4');
RAW = [RAW;raw(2:end,:)];
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet5');
RAW = [RAW;raw(2:end,:)];
raw = RAW;
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) = {''};
cellVectors = raw(:,[1,2,3,4,5,6,7,8]);
raw = raw(:,9);
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

Korvataan ei-numeeriset solut NaN-arvolla

```
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Etsitaan ei-
numeeriset solut
raw(R) = {NaN}; % Korvataan ei-numeeriset solut
```

Luodaan tulostemuuttuja

```
data = reshape([raw{:}],size(raw));
```

Nimetaan pystyvektorimuuttujat tuotuun taulukkoon

```
Aika = cellVectors(:,1);
Kokonimi1 = cellVectors(:,2);
Affecteduser2 = cellVectors(:,3);
Eventcontext2 = cellVectors(:,4);
Component2 = cellVectors(:,5);
Eventname = cellVectors(:,6);
Kuvaus2 = cellVectors(:,7);
Origin2 = cellVectors(:,8);
opnum = data(:,1);
```

Poistetaan valiaikaiset muuttujat

```
clearvars data raw cellVectors R;
```

Tuodaan datan seuraava valilehti MATLAbiin

```
[~, ~, raw] = xlsread('alkeet2.xlsx','Sheet6');
raw = raw(:,4:end);
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) = {''};
```

Korvataan ei-numeeriset solut NaN-arvolla

```
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Etsi ei-
numeeriset solut
raw(R) = {NaN}; % Korvaa ei-numeeriset solut
```

Luodaan tulostemuuttuja

```
data = reshape([raw{:}],size(raw));
```

Nimetaan pystyvektorimuuttujat tuotuun taulukkoon

```
Tunnistenumero = data(:,1);
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
T1 = data(:,2);
T2 = data(:,3);
T3 = data(:,4);
T4 = data(:,5);
T5 = data(:,6);
K = data(:,7);
Yht = data(:,8);
PTT = data(:,9);
```

Poistetaan valiaikaiset muuttujat

```
clearvars data raw R;
```

Tuodaan dataa MATLABiin tekstitiedostosta

Alustetaan muuttujat

```
filename = 'matlab-arvosanat.csv';
delimiter = ';';
```

Luetaan pystyvektorimuotoinen data merkkijonoina

```
formatSpec = '%s%s%s%[^\n\r]';
```

Avataan tekstitiedosto

```
fileID = fopen(filename,'r');
```

Luetaan pystyvektorimuotoinen data formatSpec-merkkijonomuotoilun mukaan

```
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,
  'ReturnOnError', false);
```

Suljetaan tekstitiedosto

```
fclose(fileID);
```

Muunnetaan pystyvektorien numeeriset merkkijonot numeroiksi

Korvataan ei-numeeriset merkkijonot NaN-arvoilla

```
raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
for col=1:length(dataArray)-1
    raw(1:length(dataArray{col}),col) = dataArray{col};
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3]
    % Muuntaa solutaulukon merkkijonot numeroiksi. Ei-numeeriset
    % merkkijonot korvattu NaN-arvoilla.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Luo regular expressionin ei-numeeristen alku- ja
 loppupaatteiden havaitsemiseksi.
        regexstr = '(?<prefix>.*?)(?<numbers>([-]*(\d+[\,]*)+[\.]
{0,1}\d*[eEdD]{0,1}[-+]*\d*[i]{0,1})|([-]*(\d+[\,]*)*[\.]{1,1}\d
+[eEdD]{0,1}[-+]*\d*[i]{0,1}))(?<suffix>.*)';
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Havaitsee pilkut muualla kuin tuhansien erottimena.
            invalidThousandsSeparator = false;
            if any(numbers==',');
                thousandsRegExp = '^\d+?(\,\d{3})*\.{0,1}\d*$';
                if isempty(regexp(thousandsRegExp, ',', 'once'));
                    numbers = NaN;
                    invalidThousandsSeparator = true;
                end
            end
            % Muuntaa numeeriset merkkijonot numeroiksi.
            if ~invalidThousandsSeparator;
                numbers = textscan(strrep(numbers, ',', ''), '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch me
        end
    end
end
```

Korvataan ei-numeeriset solut NaN-arvoilla

```
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Etsitaan ei-
numeeriset solut
raw(R) = {NaN}; % Korvataan ei-numeeriset solut
```

Luodaan tulostemuuttuja

```
M12 = cell2mat(raw);
```

Poistetaan valiaikaiset muuttujat

```
clearvars filename delimiter formatSpec fileID dataArray ans raw col numericData r
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

Muunnetaan aikavektori paivamaaramuotoon

```
a = Aika(1:end);
na = length(a);
at = nan(size(a));

for ii=1:na
at(ii)=datenum(a{ii},'dd.mm.yyyy HH:MM');
end

[opnumG,opnumGN,opnumGL] = grp2idx(opnum);
[Gt,GNt,GLt]=grp2idx(Eventname);
GNt(8)
```

```
ans =

    'Tenttivastaus palautettu'
```

Haetaan vain ne rivit, joissa tenttivastaus on palautettu

```
att = at(Gt==8);
idtt = opnumG(Gt==8);
iopnum = opnum(Gt==8);
numt = grpstats(att,idtt,'numel');
rt = grpstats(att,idtt,'range');
maxt = grpstats(att,idtt,'max');%Viimeinen tenttivastaus
mint = grpstats(att,idtt,'min');%Ensimmainen tenttivastaus

%Annetaan jarjestysnumero
[~,inumt]=sort(numt);
[~,irt]=sort(rt);
[~,imint]=sort(mint);
[~,imaxt]=sort(maxt);

inumt(inumt)=1:length(inumt);
irt(irt)=1:length(irt);
imint(imint)=1:length(imint);
imaxt(imaxt)=1:length(imaxt);

%Haetaan perustaitojen testin tulokset
PTTa = nan(size(opnum));

for jj=1:length(Tunnistenumero)
    idPTT = find(opnum==Tunnistenumero(jj));
    PTTa(idPTT)=PTT(jj);
end
iPTT = PTTa(Gt==8);

pttt = grpstats(iPTT,idtt,'min');
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
%Haetaan kahden ensimmaisen matematiikan kurssin arvosanat
MA1a = nan(size(opnum));
MA2a = nan(size(opnum));
for jj=1:length(M12(:,1))
    idma = find(opnum==M12(jj,1));
    MA1a(idma)=M12(jj,2);
    MA2a(idma)=M12(jj,3);
end
iMA1 = MA1a(Gt==8);

ma1 = grpstats(iMA1,idtt,'min');
iMA2 = MA2a(Gt==8);

ma2 = grpstats(iMA2,idtt,'min');
```

Tehdaan klusterianalyysi datalle

```
X = [imint,imaxt,irt];

%Tehdaan k-means -klusterointi
for ii=1:6
[clus,~,c1s] =
 kmeans(X,ii,'Distance','cityblock','Start','uniform','Display','iter');
SS(ii)=sum(c1s);
CLUS(:,ii)=clus(:);
figure; scatter(mint,maxt,clus*2+30,clus,'filled')
title(['Klustereita: ' num2str(ii) ' Kokonaissumma: '
 num2str(SS(ii))])
end
figure; plot(1:length(SS),SS,'k.-')
title('Kokonaissumma')

% Haetaan perustaitojen testin ja matematiikan kurssien arvosanojen
% keskiarvot keskihajontoineen neljan klusterin tapaukselle

disp('Opiskelijoiden matematiikan perustaitojen testin pistekeskiarvot
 keskihajontoineen klustereittain:')
meanptt4 = grpstats(pttt,CLUS(:,4),'mean')
std4 = grpstats(pttt,CLUS(:,4),'std')

disp('Opiskelijoiden ensimmaisen ja toisen matematiikan kurssin
 arvosanakeskiarvot keskihajontoineen klustereittain:')
meanma1 = grpstats(ma1,CLUS(:,4),'mean')
stdma1 = grpstats(ma1,CLUS(:,4),'std')
meanma2 = grpstats(ma2,CLUS(:,4),'mean')
stdma2 = grpstats(ma2,CLUS(:,4),'std')

% Piirretaan neljan klusterin tapauksen kuvaaja
figure; scatter(mint,maxt,CLUS(:,4)*2+30,CLUS(:,4),'filled')
grpstats(CLUS(:,4),CLUS(:,4),'numel')
hold on
meanx = grpstats(mint,CLUS(:,4),'mean')
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
meany = grpstats(maxt,CLUS(:,4),'mean')
text(meanx,meany,{'1','2','3','4'})
text(meanx+1,meany+2,num2str(meanptt4))
datetick('x','mmm dd','keeplimits')
datetick('y','mmm dd','keeplimits')

% Lasketaan t-testiarvot neljan klusterin tapaukselle
[H,P12,CI,STATS] = ttest2(pttt(CLUS(:,4)==1),pttt(CLUS(:,4)==2))
[H,P13,CI,STATS] = ttest2(pttt(CLUS(:,4)==1),pttt(CLUS(:,4)==3))
[H,P14,CI,STATS] = ttest2(pttt(CLUS(:,4)==1),pttt(CLUS(:,4)==4))
[H,P23,CI,STATS] = ttest2(pttt(CLUS(:,4)==2),pttt(CLUS(:,4)==3))
[H,P24,CI,STATS] = ttest2(pttt(CLUS(:,4)==2),pttt(CLUS(:,4)==4))
[H,P34,CI,STATS] = ttest2(pttt(CLUS(:,4)==3),pttt(CLUS(:,4)==4))



% Lasketaan alkeet suorittaneiden opiskelijoiden lukumaarat
 vuorokausissa
% ja tunneissa mitattuina
disp('Alle vuorokaudessa alkeet suorittaneiden opiskelijoiden
 lukumaara:')
length(find(rt<1))
disp('Alle tunnissa alkeet suorittaneiden opiskelijoiden lukumaara:')
length(find(rt<1/24))

% Piirretaan histogrammit opiskelijoiden suoritusaktiivisuuksista
figure
subplot(121)
hist(rt,max(rt))
axis([0,7,0,350])
title('Opiskelijoiden suoritusaktiivisuus')
ylabel('Opiskelijoiden lkm')
xlabel('vuorokautta')

subplot(122)
hist(rt,max(rt)*24)
axis([0,1,0,120])
title('Opiskelijoiden suoritusaktiivisuus')
ylabel('Opiskelijoiden lkm')
xlabel('tuntia')


% Lasketaan ja piirretaan hierarkkisen klusteroinnin kuvaajat
for ii=1:6
[clus] = clusterdata(X,'maxclust',ii,'distance','euclid');
CLUSh(:,ii)=clus(:);
figure; scatter(mint,maxt,clus*2+30,clus,'filled')
title(['Klustereita: ' num2str(ii) ' Kokonaissumma: '
 num2str(SS(ii))])
datetick('x','mmm dd','keeplimits')
datetick('y','mmm dd','keeplimits')
end

  iter  phase      num          sum
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
      1       1      696       363312
      2       2        0       363312
Best total sum of distances = 363312
   iter  phase      num          sum
      1       1      696       292992
      2       1       92       261444
      3       1       44       253530
      4       1       23       251254
      5       1       12       250625
      6       1        5       250538
      7       1        2       250528
      8       1        2       250518
      9       1        2       250502
     10       1        1       250488
     11       1        2       250472
     12       1        1       250470
     13       1        1       250465
     14       2        0       250465
Best total sum of distances = 250465
   iter  phase      num          sum
      1       1      696       231185
      2       1       58       224446
      3       1       30       222493
      4       1       22       221253
      5       1       24       220042
      6       1       21       219074
      7       1       26       217962
      8       1       24       216510
      9       1       41       213391
     10       1       52       208531
     11       1       37       205090
     12       1       23       203806
     13       1       16       203072
     14       1       11       202782
     15       1        8       202626
     16       1        5       202546
     17       1        3       202503
     18       1        1       202499
     19       1        1       202492
     20       2        0       202492
Best total sum of distances = 202492
   iter  phase      num          sum
      1       1      696       212557
      2       1      107       190193
      3       1       53       184408
      4       1       41       181852
      5       1       33       179837
      6       1       21       178934
      7       1       12       178568
      8       1       12       178243
      9       1        6       178089
     10       1        1       178088
     11       1        1       178080
     12       1        2       178077
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
    13      2        0       178074
Best total sum of distances = 178074
  iter  phase      num          sum
     1      1      696       240304
     2      1      237       184808
     3      1      106       169445
     4      1       57       164557
     5      1       55       160191
     6      1       37       157983
     7      1       29       156974
     8      1       21       156268
     9      1        9       156138
    10      1        4       156112
    11      1        2       156093
    12      1        3       156071
    13      1        4       156025
    14      1        3       155993
    15      1        3       155974
    16      1        4       155920
    17      1        2       155898
    18      1        9       155774
    19      1        2       155732
    20      1        3       155708
    21      1        1       155703
    22      1        3       155691
    23      1        3       155671
    24      1        5       155624
    25      1        6       155547
    26      1        2       155529
    27      1        2       155520
    28      1        1       155519
    29      1        1       155509
    30      2        0       155509
Best total sum of distances = 155509
  iter  phase      num          sum
     1      1      696       198921
     2      1      176       157345
     3      1       69       149867
     4      1       44       146919
     5      1       22       145499
     6      1       25       144490
     7      1       25       143743
     8      1       24       143035
     9      1       29       141820
    10      1       11       141509
    11      1       10       141315
    12      1        7       141129
    13      1        4       141075
    14      1        1       141071
    15      1        1       141067
    16      2        0       141067
Best total sum of distances = 141067
Opiskelijoiden matematiikan perustaitojen testin pistekeskiarvot
 keskihajontoineen klustereittain:
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
meanptt4 =

    7.6224
    8.5706
    8.2971
    8.2604


std4 =

    3.2149
    3.2359
    3.1793
    3.2847

Opiskelijoiden ensimmaisen ja toisen matematiikan kurssin
 arvosanakeskiarvot keskihajontoineen klustereittain:

meanma1 =

    2.1844
    2.7035
    2.0150
    2.8176


stdma1 =

    1.6018
    1.6002
    1.7100
    1.5939


meanma2 =

    2.5969
    3.0320
    2.0333
    2.9857


stdma2 =

    1.5886
    1.5808
    1.7342
    1.6182


ans =

   155
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
   183
   177
   181


meanx =

   1.0e+05 *

    7.3626
    7.3624
    7.3627
    7.3626


meany =

   1.0e+05 *

    7.3627
    7.3626
    7.3627
    7.3626


H =

     1


P12 =

    0.0100


CI =

   -1.6685
   -0.2279


STATS =

    tstat: -2.5901
       df: 311
       sd: 3.2263


H =

     0


P13 =
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

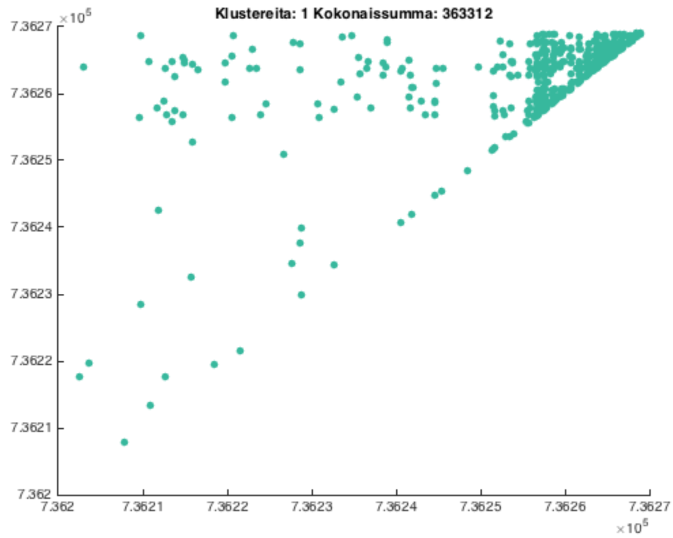
```
    0.0781


CI =

   -1.4258
    0.0764


STATS =

    tstat: -1.7684
       df: 279
       sd: 3.1975


H =

     0


P14 =

    0.0853


CI =

   -1.3652
    0.0893


STATS =

    tstat: -1.7261
       df: 310
       sd: 3.2529


H =

     0


P23 =

    0.4578


CI =

   -0.4504
    0.9974
```

LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

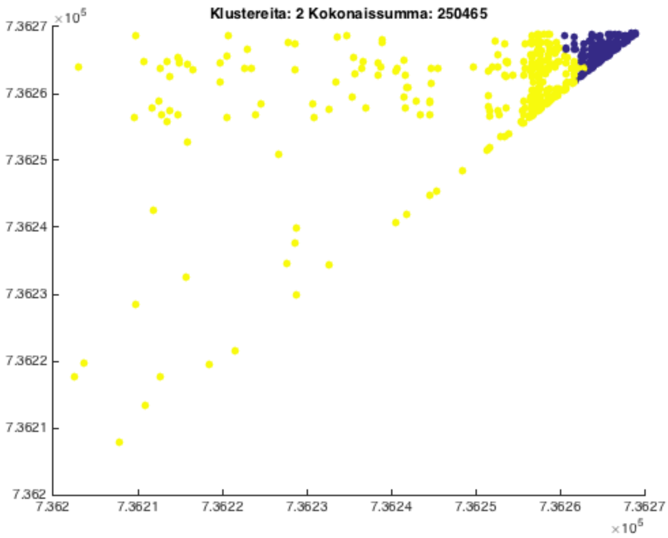
```
STATS =

    tstat: 0.7434
       df: 306
       sd: 3.2107


H =

     0


P24 =

    0.3817


CI =

   -0.3864
    1.0069


STATS =

    tstat: 0.8760
       df: 337
       sd: 3.2603


H =

     0


P34 =

    0.9213


CI =

   -0.6942
    0.7677


STATS =

    tstat: 0.0989
       df: 305
       sd: 3.2378
```

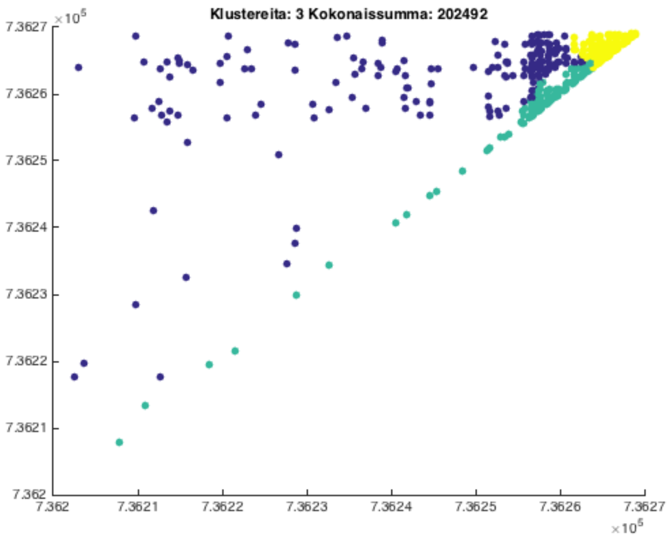
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI

```
Alle vuorokaudessa alkeet suorittaneiden opiskelijoiden lukumaara:

ans =

   338

Alle tunnissa alkeet suorittaneiden opiskelijoiden lukumaara:

ans =

   100
```

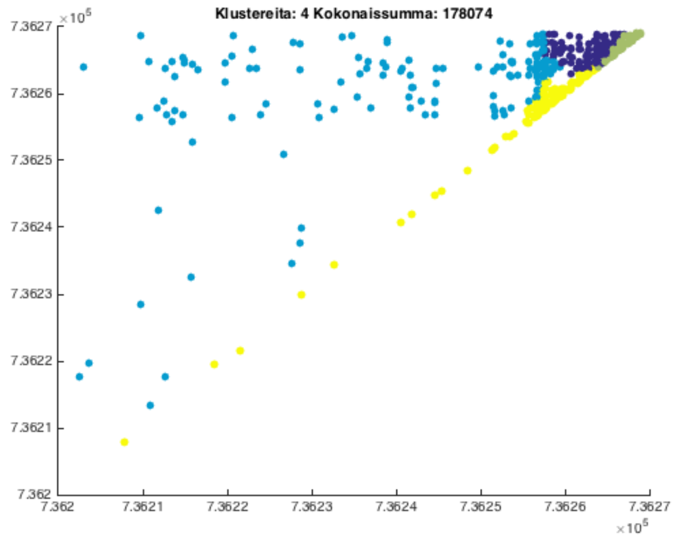


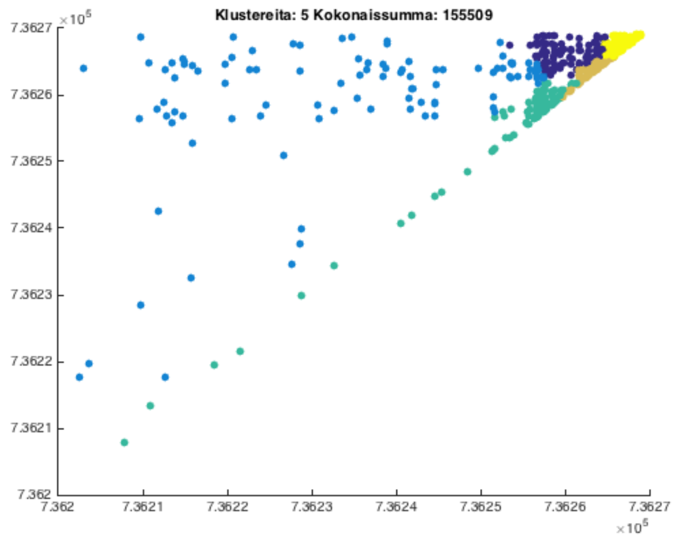
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI



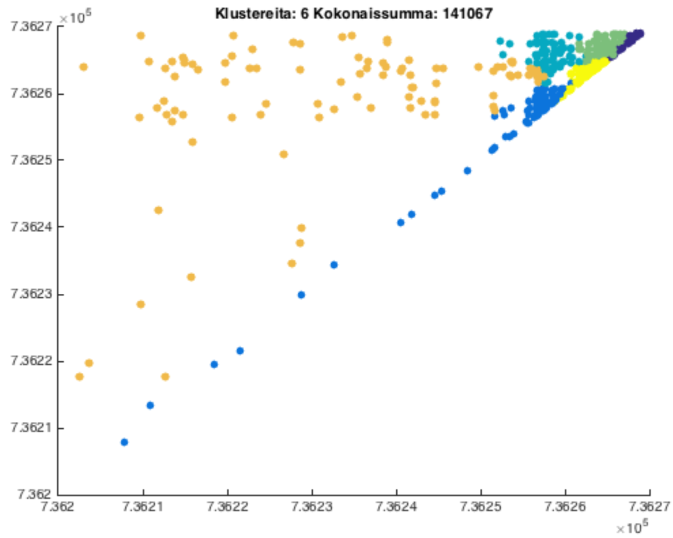


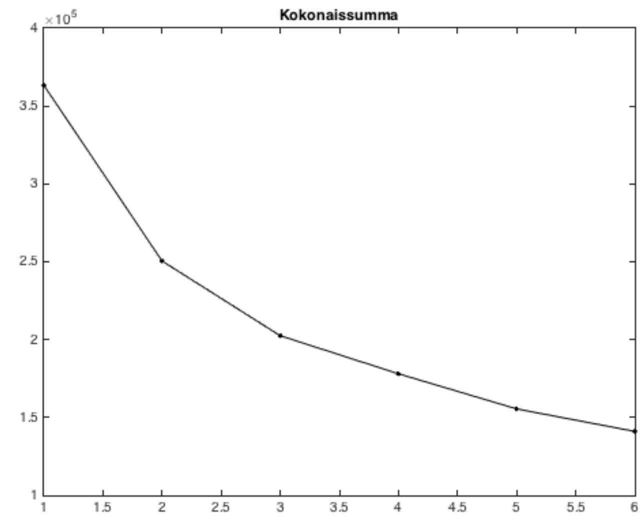
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI



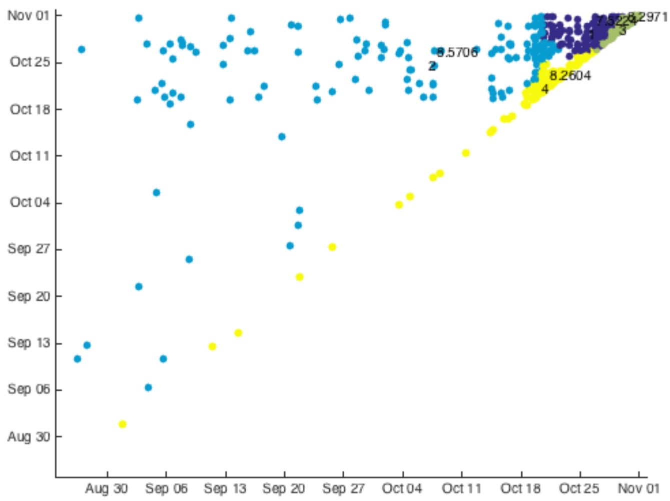


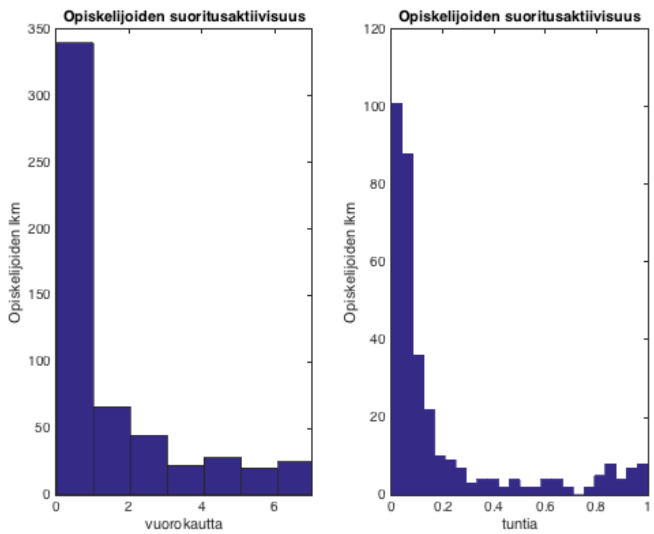
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI



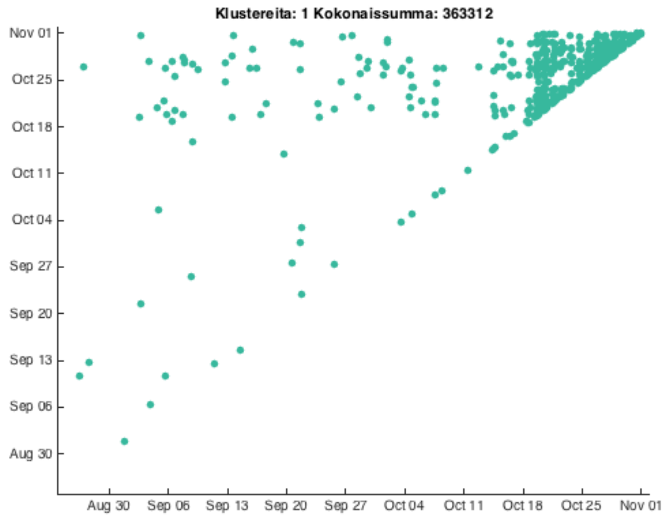


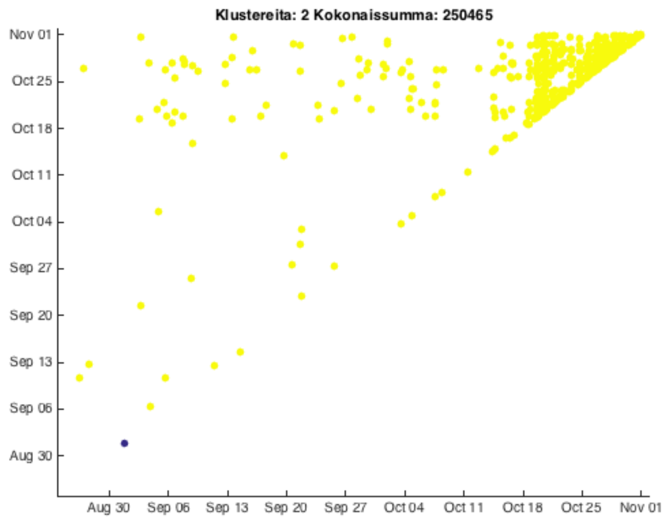
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI



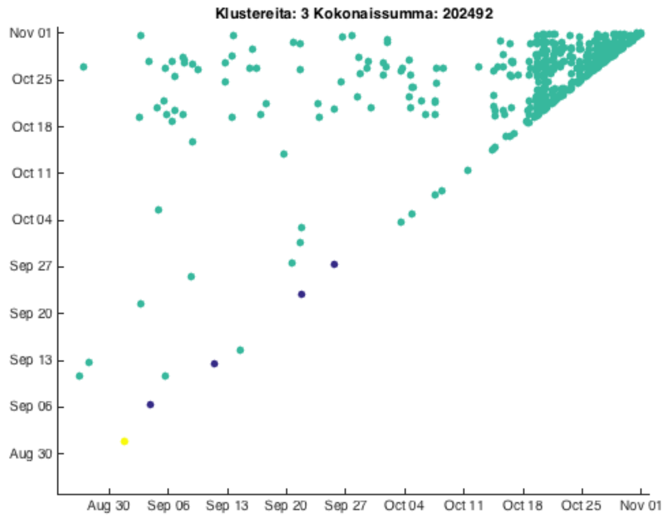


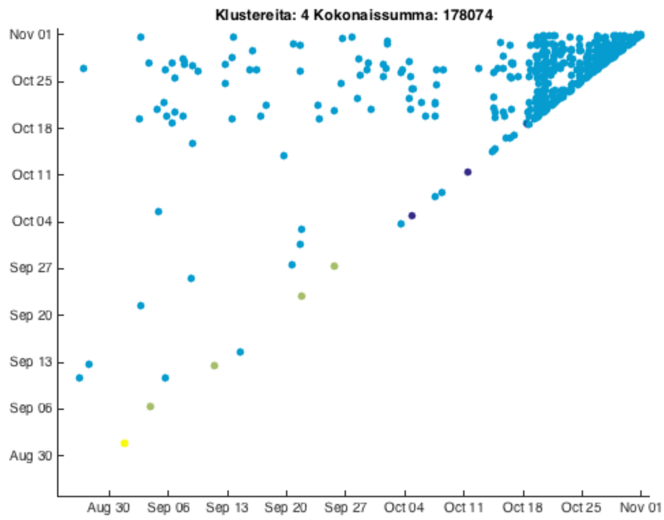
LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI





LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI





LIITE 11: KLUSTERIANALYYSIN MATLAB-KOODI